```
                                          G 12

            JJJ     000000000     BBBBBBBBBBBB        CCCCCCCCCCCC  TTTTTTTTTTTTTTT  LLL
            JJJ     000000000     BBBBBBBBBBBB        CCCCCCCCCCCC  TTTTTTTTTTTTTTT  LLL
            JJJ     000000000     BBBBBBBBBBBB        CCCCCCCCCCCC  TTTTTTTTTTTTTTT  LLL
            JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
            JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
            JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
            JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
            JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
            JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
            JJJ  000         000  BBBBBBBBBBBB     CCC                    TTT        LLL
            JJJ  000         000  BBBBBBBBBBBB     CCC                    TTT        LLL
            JJJ  000         000  BBBBBBBBBBBB     CCC                    TTT        LLL
JJJ         JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
JJJ         JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
JJJ         JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
JJJ         JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
JJJ         JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
JJJ         JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
   JJJJJJJJJ        000000000     BBBBBBBBBBBB        CCCCCCCCCCCC        TTT        LLLLLLLLLLLLLLL
   JJJJJJJJJ        000000000     BBBBBBBBBBBB        CCCCCCCCCCCC        TTT        LLLLLLLLLLLLLLL
   JJJJJJJJJ        000000000     BBBBBBBBBBBB        CCCCCCCCCCCC        TTT        LLLLLLLLLLLLLLL
```

```
_$2

Val
---
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
314
485
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
```

```
**FILE**ID**BUFFERS
```

```
BBBBBBBB    UU      UU  FFFFFFFFFF  FFFFFFFFFF  EEEEEEEEEE  RRRRRRRR      SSSSSSSS
BBBBBBBB    UU      UU  FFFFFFFFFF  FFFFFFFFFF  EEEEEEEEEE  RRRRRRRR      SSSSSSSS
BB      BB  UU      UU  FF          FF          EE          RR      RR  SS
BB      BB  UU      UU  FF          FF          EE          RR      RR  SS
BB      BB  UU      UU  FF          FF          EE          RR      RR  SS
BB      BB  UU      UU  FF          FF          EE          RR      RR  SS
BBBBBBBB    UU      UU  FFFFFFFF    FFFFFFFF    EEEEEEEE    RRRRRRRR      SSSSSS
BBBBBBBB    UU      UU  FFFFFFFF    FFFFFFFF    EEEEEEEE    RRRRRRRR      SSSSSS
BB      BB  UU      UU  FF          FF          EE          RR  RR              SS
BB      BB  UU      UU  FF          FF          EE          RR  RR              SS
BB      BB  UU      UU  FF          FF          EE          RR    RR            SS     ....
BB      BB  UU      UU  FF          FF          EE          RR    RR            SS     ....
BBBBBBBB    UUUUUUUUUU  FF          FF          EEEEEEEEEE  RR      RR  SSSSSSSS       ....
BBBBBBBB    UUUUUUUUUU  FF          FF          EEEEEEEEEE  RR      RR  SSSSSSSS       ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE BUFFERS  (%TITLE 'Buffer management utilities'
0                  IDENT = 'V04-001'
0                  ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:
1 !       Job controller.
1 !
1 ! ABSTRACT:
1 !       This module contains the buffer management utilities.
1 !
1 ! ENVIRONMENT:
1 !       VAX/VMS user and kernel mode.
1 !--
1 !
1 ! AUTHOR: M. Jack, CREATION DATE: 16-Feb-1982
1 !
1 ! MODIFIED BY:
1 !
1 !       V04-001 JAK0236         J A Krycka      14-Sep-1984
1 !               Collect more diagnostic information.
1 !
1 !       V03-008 JAK0223         J A Krycka      10-Aug-1984
1 !               Retry enqueue for queue file lock on SS$_DEADLOCK error to
1 !               prevent aborting job controller on false deadlock indication.
1 !
1 !       V03-007 JAK0219         J A Krycka      17-Jul-1984
1 !               Track changes in JOBCTLDEF.REQ.
1 !
1 !       V03-006 KPL0001         P Lieberwirth,  10-Jul-1984
1 !               Add global routine FLUSH_RECORD, to be used when a record
```

```
;    58         0058   1 !                    needs to be written back to the queue-file for reliability
;    59         0059   1 !                    reasons but is also still needed for subsequent processing.
;    60         0060   1 !                    This routine provides better performance than doing a pair
;    61         0061   1 !                    of REWRITE_RECORD/READ_RECORD calls.
;    62         0062   1 !
;    63         0063   1 !            V03-005 JAK0211           J A Krycka      14-May-1984
;    64         0064   1 !                    Continuation of V03-004.
;    65         0065   1 !
;    66         0066   1 !            V03-004 JAK0210           J A Krycka      10-May-1984
;    67         0067   1 !                    Conditionally set ROP record locking options.
;    68         0068   1 !
;    69         0069   1 !            V03-003 JAK0209           J A Krycka      09-May-1984
;    70         0070   1 !                    Log diagnostic information in DIAG_TRACE vector.
;    71         0071   1 !
;    72         0072   1 !            V03-002 MLJ0114           Martin L. Jack, 23-Jun-1983
;    73         0073   1 !                    Changes for job controller baselevel.
;    74         0074   1 !
;    75         0075   1 !            V03-001 MLJ0109           Martin L. Jack, 14-Apr-1983
;    76         0076   1 !                    Changes for job controller baselevel.
;    77         0077   1 !
;    78         0078   1 !**
;     80        0079    1 REQUIRE 'SRC$:JOBCTLDEF';
;     81        1120    1
;     82        1121    1
;     83        1122    1 FORWARD ROUTINE
;     84        1123    1         ALLOCATE_MEMORY,
;     85        1124    1         DEALLOCATE_MEMORY:                NOVALUE,
;     86        1125    1         ALLOCATE_BUFFER,
;     87        1126    1         DEALLOCATE_BUFFER:                NOVALUE,
;     88        1127    1         ALLOCATE_RECORD:                  L_OUTPUT_2,
;     89        1128    1         DEALLOCATE_RECORD:                NOVALUE,
;     90        1129    1         DEALLOCATE_RECORD_LIST:           NOVALUE,
;     91        1130    1         READ_RECORD,
;     92        1131    1         FLUSH_RECORD:                     NOVALUE,
;     93        1132    1         REWRITE_RECORD:                   NOVALUE,
;     94        1133    1         RELEASE_RECORD:                   NOVALUE,
;     95        1134    1         LOCK_QUEUE_FILE:                  NOVALUE,
;     96        1135    1         UNLOCK_QUEUE_FILE:                NOVALUE;
;     97        1136    1
;     98        1137    1
;     99        1138    1 EXTERNAL ROUTINE
;    100        1139    1         DELETE_OPEN_GETQUIS:              NOVALUE,
;    101        1140    1         SIGNAL_FILE_ERROR:                NOVALUE;
;    102        1141    1
;    103        1142    1
;    104        1143    1 ! Buffer control block.
;    105        1144    1 !
;    106        1145    1 MACRO
;    107        1146    1         BCB_LINK=             0,0,32,0 %,     ! Link to next block
;    108        1147    1         BCB_RECNO=            4,0,32,0 %,     ! Record number
;    109        1148    1         BCB_REFCOUNT=         8,0,32,0 %,     ! Reference count
;    110        1149    1         BCB_BUFFER=           12,0,32,0 %;    ! Pointer to buffer
;    111        1150    1
;    112        1151    1
;    113        1152    1 LITERAL
;    114        1153    1         BCB_K_LENGTH=         16;             ! Block length
;    115        1154    1
;    116        1155    1
;    117        1156    1 BUILTIN
;    118        1157    1         INSQUE,
;    119        1158    1         REMQUE,
;    120        1159    1         TESTBITCS,
;    121        1160    1         TESTBITSC;
```

```
1 GLOBAL ROUTINE ALLOCATE_MEMORY(PAGES)=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine allocates pages of virtual memory.
1 !
1 ! INPUT PARAMETERS:
1 !       PAGES           - (Optional) Number of pages to allocate.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       Pointer to the allocated memory.
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 LOCAL
2       P:      REF VECTOR,     ! Pointer to allocated block
2       NP;                     ! Number of pages to allocate
2 BUILTIN
2       ACTUALCOUNT;
2
2 !
2 ! Get the number of pages to allocate.
2 !
2 NP = 1;
2 IF ACTUALCOUNT() GTRU 0 THEN NP = .PAGES;
2
2
2 IF NOT REMQUE(.MEMORY_FREE_QUEUES[2*(.NP-1)], P)
2 THEN
3     BEGIN
3     P[0] = P[1] = 0;
3     .P
3     END
2 ELSE
3     BEGIN
3     LOCAL
3         RETADR:         VECTOR[2],      ! $EXPREG return address array
3         STATUS;                         ! Status return
3
3     STATUS = $EXPREG(PAGCNT=.NP, RETADR=RETADR);
3     IF NOT .STATUS THEN SIGNAL(JBC$_ALLOCMEM OR STS$K_SEVERE, 0, .STATUS);
3     .RETADR[0]
3     END
```

```
;  180          1218  1 END;

                                                   .TITLE  BUFFERS Buffer management utilities
                                                   .IDENT  \V04-001\

                                                   .PSECT  COMMON,NOEXE,  OVR,2

                                    00000 DIAG_STORAGE_BASE:
                                                   .BLKB   0
                                    00000 DIAG_TRACE:
                                                   .BLKB   96
                                    00060 DIAG_COUNT:
                                                   .BLKB   96
                                    000C0 DIAG_FLAGS:
                                                   .BLKB   4
                                    000C4 WORK_AREA:
                                                   .BLKB   44
                                    000F0 SNDJBC_COUNT:
                                                   .BLKB   132
                                    00174 GETQUI_COUNT:
                                                   .BLKB   40
                                    0019C SNDACC_COUNT:
                                                   .BLKB   28
                                    001B8 SNDSMB_COUNT:
                                                   .BLKB   72
                                    00200 DIAG_STORAGE_END:
                                                   .BLKB   0
                                    00200 FLAGS:   .BLKB   4
                                    00204 IMAGE_DUMP_STSFLG:
                                                   .BLKB   4
                                    00208 THIS_SYSID:
                                                   .BLKB   6
                                    0020E          .BLKB   2
                                    00210 CUR_TIME:
                                                   .BLKB   8
                                    00218 HOURLY_TIME:
                                                   .BLKB   8
                                    00220 HOURLY_PARAMS:
                                                   .BLKB   20
                                    00234 SYMBIONT_COUNT:
                                                   .BLKB   4
                                    00238 QUEUE_REFERENCE_COUNT:
                                                   .BLKB   4
                                    0023C MBX_MESSAGE_COUNT:
                                                   .BLKB   4
                                    00240 MBX:     .BLKB   4
                                    00244 MBX_END: .BLKB   4
                                    00248 MEMORY_FREE_QUEUES:
                                                   .BLKB   40
                                    00270 NONAST_WORK_QUEUE:
                                                   .BLKB   8
                                    00278 BCB_FREE_LIST:
                                                   .BLKB   4
                                    0027C BCB_ACTIVE_LIST:
                                                   .BLKB   4
                                    00280 GQL_FREE_LIST:
```

```
                .BLKB   4
00284 GQL_ACTIVE_LIST:
                .BLKB   4
00288 OPEN_GETQUI_LIST:
                .BLKB   4
0028C PROCESS_DATA_LIST:
                .BLKB   4
00290 SYMBIONT_CONTROL:
                .BLKB   4
00294 SPARE_AREA:
                .BLKB   12
002A0 REMOTE_REQUEST_LKSB:
                .BLKB   8
002A8 QUEUE_FILE_LKSB:
                .BLKB   8
002B0 QUEUE_LOCK_LKSB:
                .BLKB   8
002B8 RSP:      .BLKB   8
002C0 JBC_PRIORITY:
                .BLKB   4
002C4 JBC_PRIVILEGES:
                .BLKB   8
002CC JBC_QUOTAS:
                .BLKB   66
0030E           .BLKB   2
00310 JBC_UIC:  .BLKB   4
00314 QUEUE_FAB:
                .BLKB   80
00364 QUEUE_RAB:
                .BLKB   68
003A8 QUEUE_NAM:
                .BLKB   96
00408 QUEUE_XAB:
                .BLKB   88
00460 QUEUE_RSA:
                .BLKB   255
0055F           .BLKB   1
00560 QUEUE_ALQ:
                .BLKB   4
00564 QUEUE_MBF:
                .BLKB   1
00565           .BLKB   3
00568 ACCOUNTING_FABS:
                .BLKB   8
00570 ACCOUNTING_RABS:
                .BLKB   8
00578 ACCOUNT_FAB_A:
                .BLKB   80
005C8 ACCOUNT_RAB_A:
                .BLKB   68
0060C ACCOUNT_NAM_A:
                .BLKB   96
0066C ACCOUNT_RSA_A:
                .BLKB   255
0076B           .BLKB   1
0076C ACCOUNT_FAB_B:
                .BLKB   80
```

```
007BC ACCOUNT_RAB_B:
                .BLKB   68
00800 ACCOUNT_NAM_B:
                .BLKB   96
00860 ACCOUNT_RSA_B:
                .BLKB   255
0095F           .BLKB   1
00960 DIAG_FAB:
                .BLKB   80
009B0 DIAG_RAB:
                .BLKB   68
009F4 MBX_CHAN:
                .BLKB   4
009F8 MBX_IOSB:
                .BLKB   8
00A00 MBX_BUFFER:
                .BLKB   1024
00E00 VALUE_STORAGE_BASE:
                .BLKB   0
00E00 ITEM_PRESENT:
                .BLKB   32
00E20 VALUE_GETQUI_BASE:
                .BLKB   0
00E20 VALUE_ACCOUNTING_MESSAGE:
                .BLKB   6
00E26 VALUE_ACCOUNTING_TYPES:
                .BLKB   4
00E2A VALUE_AFTER_TIME:
                .BLKB   8
00E32 VALUE_ALIGNMENT_PAGES:
                .BLKB   1
00E33 VALUE_BASE_PRIORITY:
                .BLKB   1
00E34 VALUE_BATCH_INPUT:
                .BLKB   6
00E3A VALUE_BATCH_OUTPUT:
                .BLKB   10
00E44 VALUE_BUFFER_COUNT:
                .BLKB   1
00E45 VALUE_CHARACTERISTIC_NAME:
                .BLKB   6
00E4B VALUE_CHARACTERISTIC_NUMBER:
                .BLKB   1
00E4C VALUE_CHARACTERISTICS:
                .BLKB   16
00E5C VALUE_CHECKPOINT_DATA:
                .BLKB   6
00E62 VALUE_CLI:
                .BLKB   6
00E68 VALUE_CPU_DEFAULT:
                .BLKB   4
00E6C VALUE_CPU_LIMIT:
                .BLKB   4
00E70 VALUE_DESTINATION_QUEUE:
                .BLKB   8
00E78 VALUE_DEVICE_NAME:
                .BLKB   6
```

```
00E7E VALUE_ENTRY_NUMBER:
               .BLKB    4
00E82 VALUE_ENTRY_NUMBER_OUTPUT:
               .BLKB    10
00E8C VALUE_EXTEND_QUANTITY:
               .BLKB    2
00E8E VALUE_FILE_COPIES:
               .BLKB    1
00E8F VALUE_FILE_IDENTIFICATION:
               .BLKB    36
00EB3 VALUE_FILE_SETUP_MODULES:
               .BLKB    6
00EB9 VALUE_FILE_SPECIFICATION:
               .BLKB    6
00EBF VALUE_FIRST_PAGE:
               .BLKB    4
00EC3 VALUE_FORM_DESCRIPTION:
               .BLKB    6
00EC9 VALUE_FORM_LENGTH:
               .BLKB    1
00ECA VALUE_FORM_MARGIN_BOTTOM:
               .BLKB    1
00ECB VALUE_FORM_MARGIN_LEFT:
               .BLKB    2
00ECD VALUE_FORM_MARGIN_RIGHT:
               .BLKB    2
00ECF VALUE_FORM_MARGIN_TOP:
               .BLKB    1
00ED0 VALUE_FORM_NAME:
               .BLKB    6
00ED6 VALUE_FORM_NUMBER:
               .BLKB    4
00EDA VALUE_FORM:
               .BLKB    8
00EE2 VALUE_FORM_SETUP_MODULES:
               .BLKB    6
00EE8 VALUE_FORM_STOCK:
               .BLKB    6
00EEE VALUE_FORM_WIDTH:
               .BLKB    2
00EF0 VALUE_GENERIC_TARGET:
               .BLKB    996
012D4 VALUE_JOB_COPIES:
               .BLKB    1
012D5 VALUE_JOB_LIMIT:
               .BLKB    1
012D6 VALUE_JOB_NAME:
               .BLKB    6
012DC VALUE_JOB_RESET_MODULES:
               .BLKB    6
012E2 VALUE_JOB_SIZE_MAXIMUM:
               .BLKB    4
012E6 VALUE_JOB_SIZE_MINIMUM:
               .BLKB    4
012EA VALUE_JOB_STATUS_OUTPUT:
               .BLKB    10
012F4 VALUE_LAST_PAGE:
```

```
                .BLKB   4
012F8 VALUE_LIBRARY_SPECIFICATION:
                .BLKB   6
012FE VALUE_LOG_QUEUE:
                .BLKB   8
01306 VALUE_LOG_SPECIFICATION:
                .BLKB   6
0130C VALUE_NOTE:
                .BLKB   6
01312 VALUE_OPERATOR_REQUEST:
                .BLKB   6
01318 VALUE_OWNER_UIC:
                .BLKB   4
0131C VALUE_PAGE_SETUP_MODULES:
                .BLKB   6
01322 VALUE_PARAMETER_1:
                .BLKB   6
01328 VALUE_PARAMETER_2:
                .BLKB   6
0132E VALUE_PARAMETER_3:
                .BLKB   6
01334 VALUE_PARAMETER_4:
                .BLKB   6
0133A VALUE_PARAMETER_5:
                .BLKB   6
01340 VALUE_PARAMETER_6:
                .BLKB   6
01346 VALUE_PARAMETER_7:
                .BLKB   6
0134C VALUE_PARAMETER_8:
                .BLKB   6
01352 VALUE_PRIORITY:
                .BLKB   1
01353 VALUE_PROCESSOR:
                .BLKB   6
01359 VALUE_PROTECTION:
                .BLKB   4
0135D VALUE_QUEUE:
                .BLKB   6
01363 VALUE_QUEUE_FILE_SPECIFICATION:
                .BLKB   6
01369 VALUE_RELATIVE_PAGE:
                .BLKB   4
0136D VALUE_RESERVED_INPUT_1:
                .BLKB   1
0136E VALUE_RESERVED_INPUT_2:
                .BLKB   2
01370 VALUE_RESERVED_INPUT_3:
                .BLKB   4
01374 VALUE_RESERVED_INPUT_4:
                .BLKB   6
0137A VALUE_RESERVED_OUTPUT_1:
                .BLKB   10
01384 VALUE_RESERVED_OUTPUT_2:
                .BLKB   10
0138E VALUE_SEARCH_STRING:
                .BLKB   6
```

```
                                                                            01394 VALUE_SCSNODE_NAME:
                                                                                              .BLKB     6
                                                                            0139A VALUE_WSDEFAULT:
                                                                                              .BLKB     2
                                                                            0139C VALUE_WSEXTENT:
                                                                                              .BLKB     2
                                                                            0139E VALUE_WSQUOTA:
                                                                                              .BLKB     2
                                                                            013A0 VALUE_STORAGE_END:
                                                                                              .BLKB     0

                                                                                    JBC$_CLOSEOUT=            266328
                                                                                    JBC$_NOCMKRNL=            272388
                                                                                    JBC$_NOOPER=              272532
                                                                                    JBC$_NOSYSNAM=            272404
                                                                                    JBC$_OPENIN=              266392
                                                                                    JBC$_OPENOUT=             266400
                                                                                    JBC$_READERR=             266416
                                                                                    JBC$_WRITEERR=            266448
                                                                                              .EXTRN    DELETE_OPEN_GETQUIS
                                                                                              .EXTRN    SIGNAL_FILE_ERROR
                                                                                              .EXTRN    SYS$EXPREG

                                                                                              .PSECT    CODE,NOWRT,2

                                                                       000C 00000             .ENTRY    ALLOCATE_MEMORY, Save R2,R3                           ; 1161
                                                   5E                 08 C2 00002             SUBL2     #8, SP                                                ;
                                                   51                 01 D0 00005             MOVL      #1, NP                                                ; 1198
                                                                      6C 95 00008             TSTB      (AP)                                                  ; 1199
                                                                      04 13 0000A             BEQL      1$                                                    ;
                                                   51        04       AC D0 0000C             MOVL      PAGES, NP                                             ;
                                52                 51                 01 78 00010    1$:      ASHL      #1, NP, R2                                            ; 1202
                                                   53      00000000'EF42 DE 00014             MOVAL     MEMORY_FREE_QUEUES-8[R2], R3                          ;
                                                   50        00       B3 0F 0001C             REMQUE    @0(R3), P                                             ;
                                                                      03 1D 00020             BVS       2$                                                    ;
                                                                      60 7C 00022             CLRQ      (P)                                                   ; 1205
                                                                         04 00024             RET                                                             ; 1206
                                                                      7E 7C 00025    2$:      CLRQ      -(SP)                                                 ; 1214
                                                             08       AE 9F 00027             PUSHAB    RETADR                                                ;
                                                                      51 DD 0002A             PUSHL     NP                                                    ;
                                    00000000G      00                 04 FB 0002C             CALLS     #4, SYS$EXPREG                                        ;
                                                   11                 50 E8 00033             BLBS      STATUS, 3$                                            ; 1215
                                                                      50 DD 00036             PUSHL     STATUS                                                ;
                                                                      7E D4 00038             CLRL      -(SP)                                                 ;
                                                      0004840C        8F DD 0003A             PUSHL     #295948                                               ;
                                    00000000G      00                 03 FB 00040             CALLS     #3, LIB$SIGNAL                                        ;
                                                   50                 6E D0 00047    3$:      MOVL      RETADR, R0                                            ; 1216
                                                                         04 0004A             RET                                                             ; 1218

; Routine Size:  75 bytes,    Routine Base:  CODE + 0000
;  182     1219  1 GLOBAL ROUTINE DEALLOCATE_MEMORY(P,PAGES): NOVALUE=
;  183     1220  1
;  184     1221  1 !++
;  185     1222  1 !
;  186     1223  1 ! FUNCTIONAL DESCRIPTION:
;  187     1224  1 !       This routine deallocates pages of virtual memory.
;  188     1225  1 !
;  189     1226  1 ! INPUT PARAMETERS:
;  190     1227  1 !       P                   - Pointer to memory to be deallocated.
;  191     1228  1 !       PAGES               - (Optional) Number of pages to deallocate.
;  192     1229  1 !
;  193     1230  1 ! IMPLICIT INPUTS:
;  194     1231  1 !       NONE
;  195     1232  1 !
;  196     1233  1 ! OUTPUT PARAMETERS:
;  197     1234  1 !       NONE
;  198     1235  1 !
;  199     1236  1 ! IMPLICIT OUTPUTS:
;  200     1237  1 !       NONE
;  201     1238  1 !
;  202     1239  1 ! ROUTINE VALUE:
;  203     1240  1 !       NONE
;  204     1241  1 !
;  205     1242  1 ! SIDE EFFECTS:
;  206     1243  1 !       NONE
;  207     1244  1 !
;  208     1245  1 !--
;  209     1246  1
;  210     1247  2 BEGIN
;  211     1248  2 LOCAL
;  212     1249  2       NP;                     ! Number of pages to deallocate
;  213     1250  2 BUILTIN
;  214     1251  2       ACTUALCOUNT;
;  215     1252  2
;  216     1253  2
;  217     1254  2 ! Get the number of pages to deallocate.
;  218     1255  2 !
;  219     1256  2 NP = 1;
;  220     1257  2 IF ACTUALCOUNT() GTRU 1 THEN NP = .PAGES;
;  221     1258  2
;  222     1259  2
;  223     1260  2 ! Clear the pages and link them to the free list.
;  224     1261  2 !
;  225     1262  2 CH$FILL(0, .NP * 512, .P);
;  226     1263  2 INSQUE(.P, MEMORY_FREE_QUEUES[2*(.NP-1)]);
;  227     1264  1 END;



                                   007C 00000          .ENTRY  DEALLOCATE_MEMORY, Save R2,R3,R4,R5,R6       ; 1219
                    56          01  D0 00002          MOVL    #1, NP                                         ; 1256
                    01          6C  91 00005          CMPB    (AP), #1                                       ; 1257
                                04  1B 00008          BLEQU   1$                                             ;
                    56      08  AC  D0 0000A          MOVL    PAGES, NP                                      ;
          50        56          09  78 0000E 1$:      ASHL    #9, NP, R0                                     ; 1262
```

```
        50             00              6E        00  2C 00012          MOVC5   #0, (SP), #0, R0, @P                         ;
                                             04  BC     00017                                                               ;
                                       56        02  C4 00019          MULL2   #2, R6                                       ; 1263
                                       50 00000000'EF46  DE 0001C      MOVAL   MEMORY_FREE_QUEUES-8[R6], R0                 ;
                                       60        04  BC  0E 00024      INSQUE  @P, (R0)                                     ;
                                                         04 00028      RET                                                  ; 1264

; Routine Size:  41 bytes,    Routine Base:  CODE + 004B
;  229          1265  1 ROUTINE ALLOCATE_BUFFER=
;  230          1266  1
;  231          1267  1 !++
;  232          1268  1 !
;  233          1269  1 ! FUNCTIONAL DESCRIPTION:
;  234          1270  1 !       This routine allocates a buffer control block and a buffer from
;  235          1271  1 !       dynamic memory.
;  236          1272  1 !
;  237          1273  1 ! INPUT PARAMETERS:
;  238          1274  1 !       NONE
;  239          1275  1 !
;  240          1276  1 ! IMPLICIT INPUTS:
;  241          1277  1 !       NONE
;  242          1278  1 !
;  243          1279  1 ! OUTPUT PARAMETERS:
;  244          1280  1 !       NONE
;  245          1281  1 !
;  246          1282  1 ! IMPLICIT OUTPUTS:
;  247          1283  1 !       NONE
;  248          1284  1 !
;  249          1285  1 ! ROUTINE VALUE:
;  250          1286  1 !       Pointer to the buffer control block.
;  251          1287  1 !
;  252          1288  1 ! SIDE EFFECTS:
;  253          1289  1 !       NONE
;  254          1290  1 !
;  255          1291  1 !--
;  256          1292  1
;  257          1293  2 BEGIN
;  258          1294  2 LOCAL
;  259          1295  2         BCB:            REF BBLOCK;     ! Pointer to buffer control block
;  260          1296  2
;  261          1297  2
;  262          1298  2 ! If the buffer control block free list is empty, allocate a page of memory
;  263          1299  2 ! and transform it into free buffer control blocks.
;  264          1300  2 !
;  265          1301  2 IF .BCB_FREE_LIST EQL 0
;  266          1302  2 THEN
;  267          1303  3     BEGIN
;  268          1304  3     BCB_FREE_LIST = ALLOCATE_MEMORY();
;  269          1305  3     DECRA P
;  270          1306  3     FROM .BCB_FREE_LIST+512-2*BCB_K_LENGTH TO .BCB_FREE_LIST BY BCB_K_LENGTH DO
;  271          1307  3         .P = .P + BCB_K_LENGTH;
;  272          1308  2     END;
;  273          1309  2
;  274          1310  2
;  275          1311  2 ! Remove a buffer control block from the free list, allocate a buffer from
;  276          1312  2 ! dynamic memory, and initialize the buffer pointer.
;  277          1313  2 !
;  278          1314  2 BCB = .BCB_FREE_LIST;
;  279          1315  2 BCB_FREE_LIST = .BCB[BCB_LINK];
;  280          1316  2 BCB[BCB_BUFFER] = ALLOCATE_MEMORY();
;  281          1317  2
;  282          1318  2
;  283          1319  2 ! Return the buffer control block.
;  284          1320  2 !
;  285          1321  2 .BCB
;  286          1322  1 END;



                                                     000C 00000 ALLOCATE_BUFFER:
                                                                       .WORD    Save R2,R3                          ; 1265
                          53 00000000'    EF   9E 00002                 MOVAB    BCB_FREE_LIST, R3                   ;
                                          63   D5 00009                 TSTL     BCB_FREE_LIST                       ; 1301
                                          1E   12 0000B                 BNEQ     3$                                  ;
                      FF7A     CF         00   FB 0000D                 CALLS    #0, ALLOCATE_MEMORY                 ; 1304
                               63         50   D0 00012                 MOVL     R0, BCB_FREE_LIST                   ;
              50               63 000001F0 8F  C1 00015                 ADDL3    #496, BCB_FREE_LIST, P              ; 1305
                                          04   11 0001D                 BRB      2$                                  ;
                               60      10 A0   9E 0001F 1$:             MOVAB    16(R0), (P)                         ; 1307
                               50         10   C2 00023 2$:             SUBL2    #16, P                              ;
                               63         50   D1 00026                 CMPL     P, BCB_FREE_LIST                    ;
                                          F4   1E 00029                 BGEQU    1$                                  ;
                               52         63   D0 0002B 3$:             MOVL     BCB_FREE_LIST, BCB                  ; 1314
                               63         62   D0 0002E                 MOVL     (BCB), BCB_FREE_LIST                ; 1315
                      FF56     CF         00   FB 00031                 CALLS    #0, ALLOCATE_MEMORY                 ; 1316
                        0C     A2         50   D0 00036                 MOVL     R0, 12(BCB)                         ;
                               50         52   D0 0003A                 MOVL     BCB, R0                             ; 1322
                                               04 0003D                 RET                                          ;

; Routine Size:  62 bytes,    Routine Base:  CODE + 0074
;  288     1323 1 ROUTINE DEALLOCATE_BUFFER(BCB): NOVALUE=
;  289     1324 1
;  290     1325 1 !++
;  291     1326 1 !
;  292     1327 1 ! FUNCTIONAL DESCRIPTION:
;  293     1328 1 !        This routine deallocates a buffer control block and a buffer to
;  294     1329 1 !        dynamic memory.
;  295     1330 1 !
;  296     1331 1 ! INPUT PARAMETERS:
;  297     1332 1 !        BCB                 - Pointer to buffer control block.
;  298     1333 1 !
;  299     1334 1 ! IMPLICIT INPUTS:
;  300     1335 1 !        NONE
;  301     1336 1 !
;  302     1337 1 ! OUTPUT PARAMETERS:
;  303     1338 1 !        NONE
;  304     1339 1 !
;  305     1340 1 ! IMPLICIT OUTPUTS:
;  306     1341 1 !        NONE
;  307     1342 1 !
;  308     1343 1 ! ROUTINE VALUE:
;  309     1344 1 !        NONE
;  310     1345 1 !
;  311     1346 1 ! SIDE EFFECTS:
;  312     1347 1 !        NONE
;  313     1348 1 !
;  314     1349 1 !--
;  315     1350 1
;  316     1351 2 BEGIN
;  317     1352 2 MAP
;  318     1353 2         BCB:            REF BBLOCK;     ! Pointer to buffer control block
;  319     1354 2
;  320     1355 2 !
;  321     1356 2 ! Deallocate the buffer to the free list.
;  322     1357 2 !
;  323     1358 2 DEALLOCATE_MEMORY(.BCB[BCB_BUFFER]);
;  324     1359 2
;  325     1360 2 !
;  326     1361 2 ! Deallocate the buffer control block to the free list.
;  327     1362 2 !
;  328     1363 2 CH$COPY(4, BCB_FREE_LIST, 0, BCB_K_LENGTH, .BCB);
;  329     1364 2 BCB_FREE_LIST = .BCB;
;  330     1365 1 END;



                                         00FC 00000 DEALLOCATE_BUFFER:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7                  ; 1323
                         57 00000000'  EF  9E 00002        MOVAB   BCB_FREE_LIST, R7                       :
                         56        04  AC  D0 00009        MOVL    BCB, R6                                 ; 1358
                                   0C  A6  DD 0000D        PUSHL   12(R6)                                  :
                     85  AF            01  FB 00010        CALLS   #1, DEALLOCATE_MEMORY                   :
      10         00      67            04  2C 00014        MOVC5   #4, BCB_FREE_LIST, #0, #16, (R6)        ; 1363
                                       66     00019                                                        :
                         67            56  D0 0001A        MOVL    R6, BCB_FREE_LIST                       ; 1364
```

```
                                        04 0001D        RET                                         ; 1365

; Routine Size:  30 bytes,     Routine Base:  CODE + 00B2
;  332      1366  1 GLOBAL ROUTINE ALLOCATE_RECORD( ; REC_N, REC): L_OUTPUT_2=
;  333      1367  1
;  334      1368  1 !++
;  335      1369  1 !
;  336      1370  1 ! FUNCTIONAL DESCRIPTION:
;  337      1371  1 !       This routine allocates a record from the free list or by extending the
;  338      1372  1 !       queue file.
;  339      1373  1 !
;  340      1374  1 ! INPUT PARAMETERS:
;  341      1375  1 !       NONE
;  342      1376  1 !
;  343      1377  1 ! IMPLICIT INPUTS:
;  344      1378  1 !       NONE
;  345      1379  1 !
;  346      1380  1 ! OUTPUT PARAMETERS:
;  347      1381  1 !       REC_N           - Record number of record.
;  348      1382  1 !       REC             - Pointer to record.
;  349      1383  1 !
;  350      1384  1 ! IMPLICIT OUTPUTS:
;  351      1385  1 !       NONE
;  352      1386  1 !
;  353      1387  1 ! ROUTINE VALUE:
;  354      1388  1 !       Completion status.
;  355      1389  1 !
;  356      1390  1 ! SIDE EFFECTS:
;  357      1391  1 !       NONE
;  358      1392  1 !
;  359      1393  1 !--
;  360      1394  1
;  361      1395  2 BEGIN
;  362      1396  2 LOCAL
;  363      1397  2         SQH:            REF BBLOCK,     ! Pointer to SQH record
;  364      1398  2         PREV,                           ! Pointer to previous forward link
;  365      1399  2         BCB:            REF BBLOCK,     ! Pointer to buffer control block
;  366      1400  2         BUFFER:         BBLOCK[SYM$S_SYM],      ! Buffer for file extend
;  367      1401  2         FREE_N;                         ! Record number of free record
;  368      1402  2
;  369      1403  2
;  370      1404  2 ! Read the queue header.
;  371      1405  2 !
;  372      1406  2 SQH = READ_RECORD(SQH$K_RECNO);
;  373      1407  2
;  374      1408  2
;  375      1409  2 ! Establish the record number that will be allocated.
;  376      1410  2 !
;  377      1411  2 FREE_N = .SQH[SQH$L_FREE_LIST];
;  378      1412  2 IF .FREE_N EQL 0
;  379      1413  2 THEN
;  380      1414  3     BEGIN
;  381      1415  3
;  382      1416  3     ! Extend the queue file.
;  383      1417  3     !
;  384      1418  3     QUEUE_FAB[FAB$L_ALQ] = .QUEUE_ALQ;
;  385      1419  4     IF NOT $EXTEND(FAB=QUEUE_FAB)
;  386      1420  3     THEN
;  387      1421  4         BEGIN
;  388      1422  4         SIGNAL_FILE_ERROR(
;  389          1423  4                 JBC$_WRITEERR + STS$K_ERROR,
;  390          1424  4                 QUEUE_FAB, QUEUE_FAB);
;  391          1425  4             RETURN JBC$_NOQUESPACE;
;  392          1426  3             END;
;  393          1427  3
;  394          1428  3
;  395          1429  3         ! Link the newly allocated records to the free list.
;  396          1430  3         !
;  397          1431  3         CH$FILL(0, SYM$S_SYM, BUFFER);
;  398          1432  3         QUEUE_RAB[RAB$L_RBF] = BUFFER;
;  399          1433  3         IF .FLAGS[FLAGS_V_QUEUE_SHARED]
;  400          1434  3             THEN QUEUE_RAB[RAB$L_ROP] = RAB$M_UIF OR RAB$M_NLK
;  401          1435  3             ELSE QUEUE_RAB[RAB$L_ROP] = RAB$M_UIF;
;  402          1436  3         DECR REC_T
;  403          1437  3         FROM .SQH[SQH$L_HIGHEST_RECORD] + .QUEUE_FAB[FAB$L_ALQ]
;  404          1438  3         TO   .SQH[SQH$L_HIGHEST_RECORD] + 1 DO
;  405          1439  4             BEGIN
;  406          1440  4             BUFFER[SYM$L_LINK] = .FREE_N;
;  407          1441  4             QUEUE_RAB[RAB$L_KBF] = REC_T;
;  408          1442  4             DIAG_COUNT[1] = .DIAG_COUNT[1] + 1;
;  409          1443  5             IF NOT $PUT(RAB=QUEUE_RAB)
;  410          1444  4             THEN
;  411          1445  4                 SIGNAL_FILE_ERROR(
;  412          1446  4                     JBC$_WRITEERR + STS$K_SEVERE,
;  413          1447  4                     QUEUE_FAB, QUEUE_RAB);
;  414          1448  4             FREE_N = .REC_T;
;  415          1449  3             END;
;  416          1450  3         SQH[SQH$L_HIGHEST_RECORD] = .SQH[SQH$L_HIGHEST_RECORD] + .QUEUE_FAB[FAB$L_ALQ];
;  417          1451  3
;  418          1452  3
;  419          1453  3         ! Update the end of file pointer.
;  420          1454  3         !
;  421          1455  4         IF NOT $FLUSH(RAB=QUEUE_RAB)
;  422          1456  3         THEN
;  423          1457  3             SIGNAL_FILE_ERROR(
;  424          1458  3                 JBC$_WRITEERR + STS$K_SEVERE,
;  425          1459  3                 QUEUE_FAB, QUEUE_RAB);
;  426          1460  2         END;
;  427          1461  2
;  428          1462  2
;  429          1463  2     ! Search the active buffer control block list for the specified record number.
;  430          1464  2     !
;  431          1465  2     PREV = BCB_ACTIVE_LIST;
;  432          1466  2     WHILE ..PREV NEQ 0 DO
;  433          1467  3         BEGIN
;  434          1468  3         BCB = ..PREV;
;  435          1469  3
;  436          1470  3
;  437          1471  3         ! If the current record number is greater than the desired record number,
;  438          1472  3         ! the record is not in the list.
;  439          1473  3         !
;  440          1474  3         IF .BCB[BCB_RECNO] GTRU .FREE_N
;  441          1475  3         THEN
;  442          1476  3             EXITLOOP;
;  443          1477  3
;  444          1478  3
;  445          1479  3         ! If the current record number is equal to the desired record number,
;   446          1480  3          ! it indicates an internal error.
;   447          1481  3          !
;   448          1482  3          IF .BCB[BCB_RECNO] EQLU .FREE_N
;   449          1483  3          THEN
;   450          1484  3              SIGNAL(JBC$_INVBLOCK OR STS$K_SEVERE, 1, .FREE_N);
;   451          1485  3
;   452          1486  3
;   453          1487  3          ! Advance to the next buffer control block.
;   454          1488  3          !
;   455          1489  3          PREV = BCB[BCB_LINK];
;   456          1490  2          END;
;   457          1491  2
;   458          1492  2
;   459          1493  2      ! The specified record was not found.  Allocate a new buffer control block,
;   460          1494  2      ! link it into the list, and initialize it.
;   461          1495  2      !
;   462          1496  2      BCB = ALLOCATE_BUFFER();
;   463          1497  2      BCB[BCB_LINK] = ..PREV;
;   464          1498  2      .PREV = .BCB;
;   465          1499  2      BCB[BCB_RECNO] = .FREE_N;
;   466          1500  2      BCB[BCB_REFCOUNT] = 1;
;   467          1501  2
;   468          1502  2
;   469          1503  2      ! Read the free record.
;   470          1504  2      !
;   471          1505  2      QUEUE_RAB[RAB$L_KBF] = BCB[BCB_RECNO];
;   472          1506  2      IF .FLAGS[FLAGS_V_QUEUE_SHARED]
;   473          1507  2          THEN QUEUE_RAB[RAB$L_ROP] = RAB$M_NLK OR RAB$M_RRL
;   474          1508  2          ELSE QUEUE_RAB[RAB$L_ROP] = 0;
;   475          1509  2      QUEUE_RAB[RAB$L_UBF] = .BCB[BCB_BUFFER];
;   476          1510  2      DIAG_COUNT[0] = .DIAG_COUNT[0] + 1;
;   477          1511  3      IF NOT $GET(RAB=QUEUE_RAB)
;   478          1512  2      THEN
;   479          1513  2          SIGNAL_FILE_ERROR(
;   480          1514  2              JBC$_READERR + STS$K_SEVERE,
;   481          1515  2              QUEUE_FAB, QUEUE_RAB);
;   482          1516  2
;   483          1517  2
;   484          1518  2      ! Remove the record from the free list.
;   485          1519  2      !
;   486          1520  2      SQH[SQH$L_FREE_LIST] = .BBLOCK[.BCB[BCB_BUFFER], SYM$L_LINK];
;   487          1521  2      BBLOCK[.BCB[BCB_BUFFER], SYM$L_LINK] = 0;
;   488          1522  2
;   489          1523  2
;   490          1524  2      ! Rewrite the queue header.
;   491          1525  2      !
;   492          1526  2      REWRITE_RECORD(SQH$K_RECNO);
;   493          1527  2
;   494          1528  2
;   495          1529  2      ! Return the buffer address and record number.
;   496          1530  2      !
;   497          1531  2      REC_N = .FREE_N;
;   498          1532  2      REC = .BCB[BCB_BUFFER];
;   499          1533  2      DIAG_TRACE[2] = .FREE_N;                    ! ***** diagnostic info *****
;   500          1534  2      SS$_NORMAL
;   501          1535  1      END;
```

```
                                                                     .EXTRN   SYS$EXTEND, SYS$PUT
                                                                     .EXTRN   SYS$FLUSH, SYS$GET

                                                    01FC 00000       .ENTRY   ALLOCATE_RECORD, Save R2,R3,R4,R5,R6,R7,R8  ; 1366
                                  58   00000000G   EF 9E 00002       MOVAB    SIGNAL_FILE_ERROR, R8                       ;
                                  57   00000000'   EF 9E 00009       MOVAB    QUEUE_FAB, R7                               ;
                                  5E        FDFC   CE 9E 00010       MOVAB    -516(SP), SP                                ;
                                                   01 DD 00015       PUSHL    #1                                          ; 1406
                           0000V  CF               01 FB 00017       CALLS    #1, READ_RECORD                             ;
                                  5A               50 D0 0001C       MOVL     R0, SQH                                     ;
                                  56          38   AA D0 0001F       MOVL     56(SQH), FREE_N                             ; 1411
                                                   03 13 00023       BEQL     1$                                          ; 1412
                                                 00AB 31 00025       BRW      8$                                          ;
                              10  A7        024C   C7 D0 00028 1$:   MOVL     QUEUE_ALQ, QUEUE_FAB+16                     ; 1418
                                                   57 DD 0002E       PUSHL    R7                                          ; 1419
                       00000000G  00               01 FB 00030       CALLS    #1, SYS$EXTEND                              ;
                                  15               50 E8 00037       BLBS     R0, 2$                                      ;
                                                   57 DD 0003A       PUSHL    R7                                          ; 1422
                                                   57 DD 0003C       PUSHL    R7                                          ;
                                        000410D2   8F DD 0003E       PUSHL    #266450                                     ; 1423
                                  68               03 FB 00044       CALLS    #3, SIGNAL_FILE_ERROR                       ;
                                  50    00048030   8F D0 00047       MOVL     #294960, R0                                 ; 1425
                                                      04 0004E       RET                                                  ;
0200    8F          00            6E               00 2C 0004F 2$:   MOVC5    #0, (SP), #0, #512, BUFFER                  ; 1431
                                              04   AE    00056                                                            ;
                              78  A7          04   AE 9E 00058       MOVAB    BUFFER, QUEUE_RAB+40                        ; 1432
                    0A      FEEC  C7               03 E1 0005D       BBC      #3, FLAGS, 3$                               ; 1433
                              54  A7    00100010   8F D0 00063       MOVL     #1048592, QUEUE_RAB+4                       ; 1434
                                                   04 11 0006B       BRB      4$                                          ;
                              54  A7               10 D0 0006D 3$:   MOVL     #16, QUEUE_RAB+4                            ; 1435
                    50        40  AA          10   A7 C1 00071 4$:   ADDL3    QUEUE_FAB+16, 64(SQH), R0                   ; 1437
                    52        40  AA               01 C1 00077       ADDL3    #1, 64(SQH), R2                             ; 1438
                                  6E               50 D0 0007C       MOVL     R0, REC_T                                   ; 1436
                                                   2D 11 0007F       BRB      7$                                          ;
                              04  AE               56 D0 00081 5$:   MOVL     FREE_N, BUFFER                              ; 1440
                            0080  C7               6E 9E 00085       MOVAB    REC_T, QUEUE_RAB+48                         ; 1441
                                            FD50   C7 D6 0008A       INCL     DIAG_COUNT+4                                ; 1442
                                              50   A7 9F 0008E       PUSHAB   QUEUE_RAB                                   ; 1443
                       00000000G  00               01 FB 00091       CALLS    #1, SYS$PUT                                 ;
                                  0E               50 E8 00098       BLBS     R0, 6$                                      ;
                                              50   A7 9F 0009B       PUSHAB   QUEUE_RAB                                   ; 1445
                                                   57 DD 0009E       PUSHL    R7                                          ;
                                        000410D4   8F DD 000A0       PUSHL    #266452                                     ; 1446
                                  68               03 FB 000A6       CALLS    #3, SIGNAL_FILE_ERROR                       ;
                                  56               6E D0 000A9 6$:   MOVL     REC_T, FREE_N                               ; 1448
                                                   6E D7 000AC       DECL     REC_T                                       ; 1436
                                  52               6E D1 000AE 7$:   CMPL     REC_T, R2                                   ;
                                                   CE 18 000B1       BGEQ     5$                                          ;
                              40  AA          10   A7 C0 000B3       ADDL2    QUEUE_FAB+16, 64(SQH)                       ; 1450
                                              50   A7 9F 000B8       PUSHAB   QUEUE_RAB                                   ; 1455
                       00000000G  00               01 FB 000BB       CALLS    #1, SYS$FLUSH                               ;
                                  0E               50 E8 000C2       BLBS     R0, 8$                                      ;
                                              50   A7 9F 000C5       PUSHAB   QUEUE_RAB                                   ; 1457
                                                   57 DD 000C8       PUSHL    R7                                          ;
                                        000410D4   8F DD 000CA       PUSHL    #266452                                     ; 1458
```

```
                         68                03  FB 000D0        CALLS   #3, SIGNAL_FILE_ERROR         ;
                         53         FF68   C7  9E 000D3 8$:    MOVAB   BCB_ACTIVE_LIST, PREV         ; 1465
                                           63  D5 000D8 9$:    TSTL    (PREV)                        ; 1466
                                           21  13 000DA        BEQL    11$                           ;
                         52                63  D0 000DC        MOVL    (PREV), BCB                   ; 1468
                         56           04   A2  D1 000DF        CMPL    4(BCB), FREE_N                ; 1474
                                           18  1A 000E3        BGTRU   11$                           ;
                                           11  12 000E5        BNEQ    10$                           ; 1482
                                           56  DD 000E7        PUSHL   FREE_N                        ; 1484
                                           01  DD 000E9        PUSHL   #1                            ;
                                0004841C   8F  DD 000EB        PUSHL   #295964                       ;
             00000000G   00                03  FB 000F1        CALLS   #3, LIB$SIGNAL                ;
                         53                52  D0 000F8 10$:   MOVL    BCB, PREV                     ; 1489
                                           DB  11 000FB        BRB     9$                            ; 1466
                  FEA2   CF                00  FB 000FD 11$:   CALLS   #0, ALLOCATE_BUFFER           ; 1496
                         52                50  D0 00102        MOVL    R0, BCB                       ;
                         62                63  D0 00105        MOVL    (PREV), (BCB)                 ; 1497
                         63                52  D0 00108        MOVL    BCB, (PREV)                   ; 1498
                    04   A2                56  D0 0010B        MOVL    FREE_N, 4(BCB)                ; 1499
                    08   A2                01  D0 0010F        MOVL    #1, 8(BCB)                    ; 1500
                  0080   C7           04   A2  9E 00113        MOVAB   4(BCB), QUEUE_RAB+48          ; 1505
        0A        FEEC   C7                03  E1 00119        BBC     #3, FLAGS, 12$                ; 1506
                    54   A7     00100008   8F  D0 0011F        MOVL    #1048584, QUEUE_RAB+4         ; 1507
                                           03  11 00127        BRB     13$                           ;
                                      54   A7  D4 00129 12$:   CLRL    QUEUE_RAB+4                   ; 1508
                    74   A7           0C   A2  D0 0012C 13$:   MOVL    12(BCB), QUEUE_RAB+36         ; 1509
                                    FD4C   C7  D6 00131        INCL    DIAG_COUNT                    ; 1510
                                      50   A7  9F 00135        PUSHAB  QUEUE_RAB                     ; 1511
             00000000G   00                01  FB 00138        CALLS   #1, SYS$GET                   ;
                         0E                50  E8 0013F        BLBS    R0, 14$                       ;
                                      50   A7  9F 00142        PUSHAB  QUEUE_RAB                     ; 1513
                                           57  DD 00145        PUSHL   R7                            ;
                                000410B4   8F  DD 00147        PUSHL   #266420                       ; 1514
                         68                03  FB 0014D        CALLS   #3, SIGNAL_FILE_ERROR         ;
                    38   AA           0C   B2  D0 00150 14$:   MOVL    @12(BCB), 56(SQA)             ; 1520
                                      0C   B2  D4 00155        CLRL    @12(BCB)                      ; 1521
                                           01  DD 00158        PUSHL   #1                            ; 1526
                 0000V   CF                01  FB 0015A        CALLS   #1, REWRITE_RECORD            ;
                         5A                56  D0 0015F        MOVL    FREE_N, REC_N                 ; 1531
                         5B           0C   A2  D0 00162        MOVL    12(BCB), REC                  ; 1532
                  FCF4   C7                56  D0 00166        MOVL    FREE_N, DIAG_TRACE+8          ; 1533
                         50                01  D0 0016B        MOVL    #1, R0                        ; 1535
                                               04 0016E        RET                                   ;

; Routine Size:  367 bytes,    Routine Base:  CODE + 00D0
```

```
1 GLOBAL ROUTINE DEALLOCATE_RECORD(RECNO): NOVALUE=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine deallocates a specified record to the free record list.
1 !       The record may have a reference count of zero or one.
1 !
1 ! INPUT PARAMETERS:
1 !       RECNO           - Relative record number.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 LOCAL
2       SQH:            REF BBLOCK,     ! Pointer to SQH record
2       PREV,                           ! Pointer to previous forward link
2       BCB:            REF BBLOCK,     ! Pointer to buffer control block
2       BUFFER:         BBLOCK[SYM$S_SYM];      ! Temporary buffer
2
2
2 ! Read the queue header.
2 !
2 DIAG_TRACE[3] = .RECNO;                ! ***** diagnostic info *****
2 SQH = READ_RECORD(SQH$K_RECNO);
2
2
2 ! Search the active buffer control block list for the specified record number.
2 !
2 PREV = BCB_ACTIVE_LIST;
2 WHILE ..PREV NEQ 0 DO
3     BEGIN
3     BCB = ..PREV;
3
3
3     ! If the current record number is greater than the desired record number,
3     ! the record is not in the list.
3     !
3     IF .BCB[BCB_RECNO] GTRU .RECNO
3     THEN
3         EXITLOOP;
3
```

```
3
3     ! If the current record number is equal to the desired record number,
3     ! ensure that the record has a reference count of one.  If not, it
3     ! indicates an internal error.  Then, link the record to the free list
3     ! and rewrite it.
3     !
3     IF .BCB[BCB_RECNO] EQLU .RECNO
3     THEN
4         BEGIN
4         IF .BCB[BCB_REFCOUNT] NEQ 1
4         THEN
4             SIGNAL(JBC$_INVBLOCK OR STS$K_SEVERE, 1, .RECNO);
4
4
4         ! Link the record to the free list.
4         !
4         CH$COPY(4, SQH[SQH$L_FREE_LIST], 0, SYM$S_SYM, .BCB[BCB_BUFFER]);
4         SQH[SQH$L_FREE_LIST] = .RECNO;
4
4
4         ! Rewrite the record.
4         !
4         QUEUE_RAB[RAB$L_KBF] = BCB[BCB_RECNO];
4         QUEUE_RAB[RAB$L_RBF] = .BCB[BCB_BUFFER];
4         IF .FLAGS[FLAGS_V_QUEUE_SHARED]
4             THEN QUEUE_RAB[RAB$L_ROP] = RAB$M_UIF OR RAB$M_NLK
4             ELSE QUEUE_RAB[RAB$L_ROP] = RAB$M_UIF;
4         DIAG_COUNT[1] = .DIAG_COUNT[1] + 1;
5         IF NOT $PUT(RAB=QUEUE_RAB)
4         THEN
4             SIGNAL_FILE_ERROR(
4                 JBC$_WRITEERR + STS$K_SEVERE,
4                 QUEUE_FAB, QUEUE_RAB);
4
4
4         ! Unlink and release the buffer control block.
4         !
4         .PREV = .BCB[BCB_LINK];
4         DEALLOCATE_BUFFER(.BCB);
4
4
4         ! Rewrite the queue header.
4         !
4         REWRITE_RECORD(SQH$K_RECNO);
4         RETURN;
3         END;
3
3
3     ! Advance to the next buffer control block.
3     !
3     PREV = BCB[BCB_LINK];
2     END;
2
2
2 ! The specified record was not found.  Therefore, it has a reference count of
2 ! zero.  Using a scratch buffer, link the record to the free list and rewrite
2 ! the record.
```

```
;  617          1650  2 !
;  618          1651  2 CH$COPY(4, SQH[SQH$L_FREE_LIST], 0, SYM$S_SYM, BUFFER);
;  619          1652  2 SQH[SQH$L_FREE_LIST] = .RECNO;
;  620          1653  2 QUEUE_RAB[RAB$L_KBF] = RECNO;
;  621          1654  2 QUEUE_RAB[RAB$L_RBF] = BUFFER;
;  622          1655  2 IF .FLAGS[FLAGS_V_QUEUE_SHARED]
;  623          1656  2     THEN QUEUE_RAB[RAB$L_ROP] = RAB$M_UIF OR RAB$M_NLK
;  624          1657  2     ELSE QUEUE_RAB[RAB$L_ROP] = RAB$M_UIF;
;  625          1658  2 DIAG_COUNT[1] = .DIAG_COUNT[1] + 1;
;  626          1659  3 IF NOT $PUT(RAB=QUEUE_RAB)
;  627          1660  2 THEN
;  628          1661  2     SIGNAL_FILE_ERROR(
;  629          1662  2         JBC$_WRITEERR + STS$K_SEVERE,
;  630          1663  2         QUEUE_FAB, QUEUE_RAB);
;  631          1664  2
;  632          1665  2
;  633          1666  2 ! Rewrite the queue header.
;  634          1667  2 !
;  635          1668  2 REWRITE_RECORD(SQH$K_RECNO);
;  636          1669  1 END;



                                                 0FFC 00000           .ENTRY  DEALLOCATE_RECORD, Save R2,R3,R4,R5,R6,R7,- ; 1536
                                                                              R8,R9,R10,R11                               ;
                            5B 00000000G     00    9E 00002           MOVAB   SYS$PUT, R11                                ;
                            5A 00000000'     EF    9E 00009           MOVAB   QUEUE_RAB+4, R10                            ;
                            5E         FE00  CE    9E 00010           MOVAB   -512(SP), SP                                ;
                            59           04  AC    D0 00015           MOVL    RECNO, R9                                   ; 1574
                      FCA4  CA               59    D0 00019           MOVL    R9, DIAG_TRACE+12                           ;
                                             01    DD 0001E           PUSHL   #1                                          ; 1575
                     0000V  CF               01    FB 00020           CALLS   #1, READ_RECORD                             ;
                            57               50    D0 00025           MOVL    R0, SQH                                     ;
                            58         FF14  CA    9E 00028           MOVAB   BCB_ACTIVE_LIST, PREV                       ; 1580
                                             68    D5 0002D 1$:       TSTL    (PREV)                                      ; 1581
                                             7E    13 0002F           BEQL    7$                                          ;
                            56               68    D0 00031           MOVL    (PREV), BCB                                 ; 1583
                            59           04  A6    D1 00034           CMPL    4(BCB), R9                                  ; 1589
                                             75    1A 00038           BGTRU   7$                                          ;
                                             6D    12 0003A           BNEQ    6$                                          ; 1599
                            01           08  A6    D1 0003C           CMPL    8(BCB), #1                                  ; 1602
                                             11    13 00040           BEQL    2$                                          ;
                                             59    DD 00042           PUSHL   R9                                          ; 1604
                                             01    DD 00044           PUSHL   #1                                          ;
                                   0004841C  8F    DD 00046           PUSHL   #295964                                     ;
                   00000000G  00             03    FB 0004C           CALLS   #3, LIB$SIGNAL                              ;
 0200    8F         00         38  A7        04    2C 00053 2$:       MOVC5   #4, 56(SQH), #0, #512, @12(BCB)             ; 1609
                                         0C  B6       0005B                                                       ;
                               38  A7        59    D0 0005D           MOVL    R9, 56(SQH)                                 ; 1610
                               2C  AA        04  A6    9E 00061       MOVAB   4(BCB), QUEUE_RAB+48                        ; 1615
                               24  AA        0C  A6    D0 00066       MOVL    12(BCB), QUEUE_RAB+40                       ; 1616
                    09       FE98  CA            03    E1 0006B       BBC     #3, FLAGS, 3$                               ; 1617
                            6A 00100010          8F    D0 00071       MOVL    #1048592, QUEUE_RAB+4                       ; 1618
                                                 03    11 00078       BRB     4$                                          ;
                            6A                   10    D0 0007A 3$:   MOVL    #16, QUEUE_RAB+4                            ; 1619
```

```
                                      FCFC   CA  D6 0007D 4$:    INCL     DIAG_COUNT+4                            ; 1620
                                        FC   AA  9F 00081        PUSHAB   QUEUE_RAB                               ; 1621
                              6B             01  FB 00084        CALLS    #1, SYS$PUT                             ;
                              13             50  E8 00087        BLBS     R0, 5$                                  ;
                                        FC   AA  9F 0008A        PUSHAB   QUEUE_RAB                               ; 1623
                                        AC   AA  9F 0008D        PUSHAB   QUEUE_FAB                               ;
                                  000410D4   8F  DD 00090        PUSHL    #266452                                 ; 1624
                    00000000G EF             03  FB 00096        CALLS    #3, SIGNAL_FILE_ERROR                   ;
                              68             66  D0 0009D 5$:    MOVL     (BCB), (PREV)                           ; 1630
                                             56  DD 000A0        PUSHL    BCB                                     ; 1631
                         FDCC CF             01  FB 000A2        CALLS    #1, DEALLOCATE_BUFFER                   ;
                                             4E  11 000A7        BRB      10$                                     ; 1636
                              58             56  D0 000A9 6$:    MOVL     BCB, PREV                               ; 1643
                                           FF7E  31 000AC        BRW      1$                                      ; 1581
  0200   8F           00       38 A7             04  2C 000AF 7$:    MOVC5    #4, 56(SQH), #0, #512, BUFFER           ; 1651
                                             6E     000B7                                                         ;
                           38 A7             59  D0 000B8        MOVL     R9, 56(SQH)                             ; 1652
                           2C AA        04   AC  9E 000BC        MOVAB    RECNO, QUEUE_RAB+48                     ; 1653
                           24 AA             6E  9E 000C1        MOVAB    BUFFER, QUEUE_RAB+40                    ; 1654
                 09      FE98 CA             03  E1 000C5        BRC      #3, FLAGS, 8$                           ; 1655
                              6A 00100010    8F  D0 000CB        MOVL     #1048592, QUEUE_RAB+4                   ; 1656
                                             03  11 000D2        BRB      9$                                      ;
                              6A             10  D0 000D4 8$:    MOVL     #16, QUEUE_RAB+4                        ; 1657
                                      FCFC   CA  D6 000D7 9$:    INCL     DIAG_COUNT+4                            ; 1658
                                        FC   AA  9F 000DB        PUSHAB   QUEUE_RAB                               ; 1659
                              6B             01  FB 000DE        CALLS    #1, SYS$PUT                             ;
                              13             50  E8 000E1        BLBS     R0, 10$                                 ;
                                        FC   AA  9F 000E4        PUSHAB   QUEUE_RAB                               ; 1661
                                        AC   AA  9F 000E7        PUSHAB   QUEUE_FAB                               ;
                                  000410D4   8F  DD 000EA        PUSHL    #266452                                 ; 1662
                    00000000G EF             03  FB 000F0        CALLS    #3, SIGNAL_FILE_ERROR                   ;
                                             01  DD 000F7 10$:   PUSHL    #1                                      ; 1668
                         0000V CF            01  FB 000F9        CALLS    #1, REWRITE_RECORD                      ;
                                                 04 000FE        RET                                              ; 1669

; Routine Size:  255 bytes,    Routine Base:  CODE + 023F
```

```
1 GLOBAL ROUTINE DEALLOCATE_RECORD_LIST(RECNO): NOVALUE=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine deallocates a list of records linked through SYM$L_LINK
1 !       to the free list.
1 !
1 ! INPUT PARAMETERS:
1 !       RECNO           - Relative record number of the first record.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 LOCAL
2       LRECNO,                         ! Current record number
2       NRECNO,                         ! Next record number
2       REC:            REF BBLOCK;     ! Pointer to record
2
2
2 IF .RECNO NEQ 0
2 THEN
3     BEGIN
3
3     ! Read the queue header to avoid thrashing it.
3     !
3     READ_RECORD(SQH$K_RECNO);
3
3
3     ! Loop for all records in the list.
3     !
3     LRECNO = .RECNO;
3     WHILE .LRECNO NEQ 0 DO
4         BEGIN
4
4         ! Read the record, save the link to the next record, deallocate the
4         ! record, and advance to the next.
4         !
4         REC = READ_RECORD(.LRECNO);
4         NRECNO = .REC[SYM$L_LINK];
4         DEALLOCATE_RECORD(.LRECNO);
4         LRECNO = .NRECNO;
```

```
;  695        1727  3           END;
;  696        1728  3
;  697        1729  3
;  698        1730  3        ! Release the queue header.
;  699        1731  3        !
;  700        1732  3        RELEASE_RECORD(SQH$K_RECNO);
;  701        1733  2        END;
;  702        1734  1 END;



                                    001C 00000         .ENTRY  DEALLOCATE_RECORD_LIST, Save R2,R3,R4          ; 1670
                        04     AC   D5 00002           TSTL    RECNO                                          ; 1705
                               2D   13 00005           BEQL    3$                                             ;
                               01   DD 00007           PUSHL   #1                                             ; 1711
             0000V  CF         01   FB 00009           CALLS   #1, READ_RECORD                                ;
                    52  04     AC   D0 0000E           MOVL    RECNO, LRECNO                                  ; 1716
                               19   13 00012 1$:       BEQL    2$                                             ; 1717
                               52   DD 00014           PUSHL   LRECNO                                         ; 1723
             0000V  CF         01   FB 00016           CALLS   #1, READ_RECORD                                ;
                    54         50   D0 0001B           MOVL    R0, REC                                        ;
                    53         64   D0 0001E           MOVL    (REC), NRECNO                                  ; 1724
                               52   DD 00021           PUSHL   LRECNO                                         ; 1725
             FED9   CF         01   FB 00023           CALLS   #1, DEALLOCATE_RECORD                          ;
                    52         53   D0 00028           MOVL    NRECNO, LRECNO                                 ; 1726
                               E5   11 0002B           BRB     1$                                             ; 1717
                               01   DD 0002D 2$:       PUSHL   #1                                             ; 1732
             0000V  CF         01   FB 0002F           CALLS   #1, RELEASE_RECORD                             ;
                                    04 00034 3$:       RET                                                    ; 1734

; Routine Size:  53 bytes,    Routine Base:  CODE + 033E
```

```
1 GLOBAL ROUTINE READ_RECORD(RECNO)=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine reads a specified record.
1 !
1 ! INPUT PARAMETERS:
1 !       RECNO           - Relative record number.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       Address of buffer containing specified record.
1 !
1 ! SIDE EFFECTS:
1 !       Specified record read into buffer.
1 !
1 !--
1
2 BEGIN
2 LOCAL
2       PREV,                           ! Pointer to previous forward link
2       BCB:            REF BBLOCK;     ! Pointer to buffer control block
2
2 !
2 ! Search the active buffer control block list for the specified record number.
2 !
2 DIAG_TRACE[4] = .RECNO;         ! ***** diagnostic info *****
2 DIAG_COUNT[2] = .DIAG_COUNT[2] + 1;
2 PREV = BCB_ACTIVE_LIST;
2 WHILE ..PREV NEQ 0 DO
3     BEGIN
3     BCB = ..PREV;
3
3     !
3     ! If the current record number is greater than the desired record number,
3     ! the record is not in the list.
3     !
3     IF .BCB[BCB_RECNO] GTRU .RECNO
3     THEN
3         EXITLOOP;
3
3     !
3     ! If the current record number is equal to the desired record number,
3     ! increase the reference count on the record and return its buffer address.
3     !
3     IF .BCB[BCB_RECNO] EQLU .RECNO
3     THEN
4         BEGIN
```

```
;  761        1792  4          DIAG_COUNT[3] = .DIAG_COUNT[3] + 1;              ! We saved a read
;  762        1793  4          BCB[BCB_REFCOUNT] = .BCB[BCB_REFCOUNT] + 1;
;  763        1794  4          RETURN .BCB[BCB_BUFFER];
;  764        1795  3          END;
;  765        1796  3
;  766        1797  3
;  767        1798  3      ! Advance to the next buffer control block.
;  768        1799  3      !
;  769        1800  3      PREV = BCB[BCB_LINK];
;  770        1801  2      END;
;  771        1802  2
;  772        1803  2
;  773        1804  2  ! The specified record was not found.  Allocate a new buffer control block,
;  774        1805  2  ! link it into the list, and initialize it.
;  775        1806  2  !
;  776        1807  2  BCB = ALLOCATE_BUFFER();
;  777        1808  2  BCB[BCB_LINK] = ..PREV;
;  778        1809  2  .PREV = .BCB;
;  779        1810  2  BCB[BCB_RECNO] = .RECNO;
;  780        1811  2  BCB[BCB_REFCOUNT] = 1;
;  781        1812  2
;  782        1813  2
;  783        1814  2  ! Initialize the RAB and read the specified record.
;  784        1815  2  !
;  785        1816  2  QUEUE_RAB[RAB$L_KBF] = BCB[BCB_RECNO];
;  786        1817  2  IF .FLAGS[FLAGS_V_QUEUE_SHARED]
;  787        1818  2      THEN QUEUE_RAB[RAB$L_ROP] = RAB$M_NLK OR RAB$M_RRL
;  788        1819  2      ELSE QUEUE_RAB[RAB$L_ROP] = 0;
;  789        1820  2  QUEUE_RAB[RAB$L_UBF] = .BCB[BCB_BUFFER];
;  790        1821  2  DIAG_COUNT[0] = .DIAG_COUNT[0] + 1;
;  791        1822  3  IF NOT $GET(RAB=QUEUE_RAB)
;  792        1823  2  THEN
;  793        1824  2      SIGNAL_FILE_ERROR(
;  794        1825  2          JBC$_READERR + STS$K_SEVERE,
;  795        1826  2          QUEUE_FAB, QUEUE_RAB);
;  796        1827  2
;  797        1828  2
;  798        1829  2  ! Return the address of the record.
;  799        1830  2  !
;  800        1831  2  .BCB[BCB_BUFFER]
;  801        1832  1  END;



                                       001C 00000          .ENTRY  READ_RECORD, Save R2,R3,R4                ; 1735
                   54 00000000'   EF    9E 00002          MOVAB   QUEUE_RAB+4, R4                            ;
           FCA8    C4       04    AC    D0 00009          MOVL    RECNO, DIAG_TRACE+16                       ; 1770
                          FD00    C4    D6 0000F          INCL    DIAG_COUNT+8                               ; 1771
                   53     FF14    C4    9E 00013          MOVAB   BCB_ACTIVE_LIST, PREV                      ; 1772
                                  63    D5 00018 1$:      TSTL    (PREV)                                     ; 1773
                                  1A    13 0001A          BEQL    3$                                         ;
                   52             63    D0 0001C          MOVL    (PREV), BCB                                ; 1775
             04    AC       04    A2    D1 0001F          CMPL    4(BCB), RECNO                              ; 1781
                                  10    1A 00024          BGTRU   3$                                         ;
                                  09    12 00026          BNEQ    2$                                         ; 1789
```

```
                                              FD04   C4  D6 00028         INCL    DIAG_COUNT+12                                ; 1792
                                                08   A2  D6 0002C         INCL    8(BCB)                                       ; 1793
                                                     5B  11 0002F         BRB     6$                                           ; 1794
                                       53            52  D0 00031 2$:     MOVL    BCB, PREV                                    ; 1800
                                                     E2  11 00034         BRB     1$                                           ; 1773
                                FCC6   CF            00  FB 00036 3$:     CALLS   #0, ALLOCATE_BUFFER                          ; 1807
                                       52            50  D0 0003B         MOVL    R0, BCB                                      ;
                                       62            63  D0 0003E         MOVL    (PREV), (BCB)                                ; 1808
                                       63            52  D0 00041         MOVL    BCB, (PREV)                                  ; 1809
                                  04   A2         04 AC  D0 00044         MOVL    RECNO, 4(BCB)                                ; 1810
                                  08   A2            01  D0 00049         MOVL    #1, 8(BCB)                                   ; 1811
                                  2C   A4         04 A2  9E 0004D         MOVAB   4(BCB), QUEUE_RAB+48                         ; 1816
                         09     FE98   C4            03  E1 00052         BBC     #3, FLAGS, 4$                                ; 1817
                                       64 00100008   8F  D0 00058         MOVL    #1048584, QUEUE_RAB+4                        ; 1818
                                                     02  11 0005F         BRB     5$                                           ;
                                                     64  D4 00061 4$:     CLRL    QUEUE_RAB+4                                  ; 1819
                                  20   A4         0C A2  D0 00063 5$:     MOVL    12(BCB), QUEUE_RAB+36                        ; 1820
                                              FCF8   C4  D6 00068         INCL    DIAG_COUNT                                   ; 1821
                                                FC   A4  9F 0006C         PUSHAB  QUEUE_RAB                                    ; 1822
                            00000000G  00            01  FB 0006F         CALLS   #1, SYS$GET                                  ;
                                       13            50  E8 00076         BLBS    R0, 6$                                       ;
                                                FC   A4  9F 00079         PUSHAB  QUEUE_RAB                                    ; 1824
                                                AC   A4  9F 0007C         PUSHAB  QUEUE_FAB                                    ;
                                          000410B4   8F  DD 0007F         PUSHL   #266420                                      ; 1825
                            00000000G  EF            03  FB 00085         CALLS   #3, SIGNAL_FILE_ERROR                        ;
                                       50         0C A2  D0 0008C 6$:     MOVL    12(BCB), R0                                  ; 1832
                                                         04 00090         RET                                                  ;

; Routine Size:  145 bytes,    Routine Base:  CODE + 0373
```

```
1 GLOBAL ROUTINE FLUSH_RECORD(RECNO): NOVALUE=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine flushes a specified record to the queue file without
1 !       releasing access to it.
1 !
1 ! INPUT PARAMETERS:
1 !       RECNO           - Relative record number.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE·
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 LOCAL
2       PREV,                           ! Pointer to previous forward link
2       BCB:            REF BBLOCK;     ! Pointer to buffer control block
2
2
2 ! Search the active buffer control block list for the specified record number.
2 !
2 DIAG_TRACE[6] = .RECNO;         ! ***** diagnostic info *****
2 DIAG_COUNT[5] = .DIAG_COUNT[5] + 1;
2 PREV = BCB_ACTIVE_LIST;
2 WHILE ..PREV NEQ 0 DO
3     BEGIN
3     BCB = ..PREV;
3
3
3     ! If the current record number is greater than the desired record number,
3     ! the record is not in the list.
3     !
3     IF .BCB[BCB_RECNO] GTRU .RECNO
3     THEN
3         EXITLOOP;
3
3
3     ! If the current record number is equal to the desired record number,
3     ! flush the record.
3     !
3     IF .BCB[BCB_RECNO] EQLU .RECNO
3     THEN
```

```
;  860          1890  4            BEGIN
;  861          1891  4
;  862          1892  4            ! Rewrite the record.
;  863          1893  4            !
;  864          1894  4            QUEUE_RAB[RAB$L_KBF] = BCB[BCB_RECNO];
;  865          1895  4            QUEUE_RAB[RAB$L_RBF] = .BCB[BCB_BUFFER];
;  866          1896  4            IF .FLAGS[FLAGS_V_QUEUE_SHARED]
;  867          1897  4                THEN QUEUE_RAB[RAB$L_ROP] = RAB$M_UIF OR RAB$M_NLK
;  868          1898  4                ELSE QUEUE_RAB[RAB$L_ROP] = RAB$M_UIF;
;  869          1899  4            DIAG_COUNT[1] = .DIAG_COUNT[1] + 1;
;  870          1900  5            IF NOT $PUT(RAB=QUEUE_RAB)
;  871          1901  4            THEN
;  872          1902  4                SIGNAL_FILE_ERROR(
;  873          1903  4                    JBC$_WRITEERR + STS$K_SEVERE,
;  874          1904  4                    QUEUE_FAB, QUEUE_RAB);
;  875          1905  4
;  876          1906  4            RETURN;
;  877          1907  3            END;
;  878          1908  3
;  879          1909  3
;  880          1910  3        ! Advance to the next buffer control block.
;  881          1911  3        !
;  882          1912  3        PREV = BCB[BCB_LINK];
;  883          1913  2        END;
;  884          1914  2
;  885          1915  2
;  886          1916  2  ! The specified record was not found.  This condition indicates an internal
;  887          1917  2  ! logic error.
;  888          1918  2  !
;  889          1919  2  SIGNAL(JBC$_INVBLOCK OR STS$K_SEVERE, 1, .RECNO);
;  890          1920  1  END;



                                           001C 00000          .ENTRY  FLUSH_RECORD, Save R2,R3,R4                  ; 1833
                  54 00000000'       EF    9E 00002          MOVAB   QUEUE_RAB+4, R4                                ;
          FCB0    C4          04     AC    D0 00009          MOVL    RECNO, DIAG_TRACE+24                           ; 1869
                            FD0C     C4    D6 0000F          INCL    DIAG_COUNT+20                                  ; 1870
                  53        FF14     C4    9E 00013          MOVAB   BCB_ACTIVE_LIST, PREV                          ; 1871
                                     63    D5 00018 1$:      TSTL    (PREV)                                         ; 1872
                                     52    13 0001A          BEQL    5$                                             ;
                  52                 63    D0 0001C          MOVL    (PREV), BCB                                    ; 1874
            04    AC          04     A2    D1 0001F          CMPL    4(BCB), RECNO                                  ; 1880
                                     48    1A 00024          BGTRU   5$                                             ;
                                     41    12 00026          BNEQ    4$                                             ; 1888
                  2C    A4    04     A2    9E 00028          MOVAB   4(BCB), QUEUE_RAB+48                           ; 1894
                  24    A4    0C     A2    D0 0002D          MOVL    12(BCB), QUEUE_RAB+40                          ; 1895
      09    FE98    C4               03    E1 00032          BBC     #3, FLAGS, 2$                                  ; 1896
                  64 00100010        8F    D0 00038          MOVL    #1048592, QUEUE_RAB+4                          ; 1897
                                     03    11 0003F          BRB     3$                                             ;
                  64                 10    D0 00041 2$:      MOVL    #16, QUEUE_RAB+4                               ; 1898
                            FCFC     C4    D6 00044 3$:      INCL    DIAG_COUNT+4                                   ; 1899
                              FC     A4    9F 00048          PUSHAB  QUEUE_RAB                                      ; 1900
        00000000G   00               01    FB 0004B          CALLS   #1, SYS$PUT                                    ;
                    2B               50    E8 00052          BLBS    R0, 6$                                         ;
```

```
                         FC    A4  9F 00055          PUSHAB  QUEUE_RAB                            ; 1902
                         AC    A4  9F 00058          PUSHAB  QUEUE_FAB                            ;
                   000410D4    8F  DD 0005B          PUSHL   #266452                              ; 1903
        00000000G  EF          03  FB 00061          CALLS   #3, SIGNAL_FILE_ERROR                ;
                                   04 00068          RET                                          ; 1890
                  53           52  D0 00069 4$:      MOVL    BCB, PREV                            ; 1912
                               AA  11 0006C          BRB     1$                                   ; 1872
                         04    AC  DD 0006E 5$:      PUSHL   RECNO                                ; 1919
                               01  DD 00071          PUSHL   #1                                   ;
                   0004841C    8F  DD 00073          PUSHL   #295964                              ;
        00000000G  00          03  FB 00079          CALLS   #3, LIB$SIGNAL                       ;
                                   04 00080 6$:      RET                                          ; 1920

; Routine Size:  129 bytes,    Routine Base:  CODE + 0404
```

```
1 GLOBAL ROUTINE REWRITE_RECORD(RECNO): NOVALUE=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine rewrites a specified record and decreases its reference
1 !       count.
1 !
1 ! INPUT PARAMETERS:
1 !       RECNO           - Relative record number.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 LOCAL
2       PREV,                                   ! Pointer to previous forward link
2       BCB:            REF BBLOCK;             ! Pointer to buffer control block
2
2
2 ! Search the active buffer control block list for the specified record number.
2 !
2 DIAG_TRACE[5] = .RECNO;               ! ***** diagnostic info *****
2 DIAG_COUNT[4] = .DIAG_COUNT[4] + 1;
2 PREV = BCB_ACTIVE_LIST;
2 WHILE ..PREV NEQ 0 DO
3     BEGIN
3     BCB = ..PREV;
3
3
3     ! If the current record number is greater than the desired record number,
3     ! the record is not in the list.
3     !
3     IF .BCB[BCB_RECNO] GTRU .RECNO
3     THEN
3         EXITLOOP;
3
3
3     ! If the current record number is equal to the desired record number,
3     ! decrease the reference count on the record, and rewrite the record.
3     ! If the reference count has fallen to zero, deallocate the buffer
3     ! control block.
3     !
```

```
;  949          1978  3          IF .BCB[BCB_RECNO] EQLU .RECNO
;  950          1979  3          THEN
;  951          1980  4              BEGIN
;  952          1981  4
;  953          1982  4              ! Rewrite the record.
;  954          1983  4              !
;  955          1984  4              QUEUE_RAB[RAB$L_KBF] = BCB[BCB_RECNO];
;  956          1985  4              QUEUE_RAB[RAB$L_RBF] = .BCB[BCB_BUFFER];
;  957          1986  4              IF .FLAGS[FLAGS_V_QUEUE_SHARED]
;  958          1987  4                  THEN QUEUE_RAB[RAB$L_ROP] = RAB$M_UIF OR RAB$M_NLK
;  959          1988  4                  ELSE QUEUE_RAB[RAB$L_ROP] = RAB$M_UIF;
;  960          1989  4              DIAG_COUNT[1] = .DIAG_COUNT[1] + 1;
;  961          1990  5              IF NOT $PUT(RAB=QUEUE_RAB)
;  962          1991  4              THEN
;  963          1992  4                  SIGNAL_FILE_ERROR(
;  964          1993  4                      JBC$_WRITEERR + STS$K_SEVERE,
;  965          1994  4                      QUEUE_FAB, QUEUE_RAB);
;  966          1995  4
;  967          1996  4
;  968          1997  4              ! If the reference count has fallen to zero, unlink and release the
;  969          1998  4              ! buffer control block.
;  970          1999  4              !
;  971          2000  4              BCB[BCB_REFCOUNT] = .BCB[BCB_REFCOUNT] - 1;
;  972          2001  4              IF .BCB[BCB_REFCOUNT] EQL 0
;  973          2002  4              THEN
;  974          2003  5                  BEGIN
;  975          2004  5                  .PREV = .BCB[BCB_LINK];
;  976          2005  5                  DEALLOCATE_BUFFER(.BCB);
;  977          2006  4                  END;
;  978          2007  4              RETURN;
;  979          2008  3              END;
;  980          2009  3
;  981          2010  3
;  982          2011  3          ! Advance to the next buffer control block.
;  983          2012  3          !
;  984          2013  3          PREV = BCB[BCB_LINK];
;  985          2014  2          END;
;  986          2015  2
;  987          2016  2
;  988          2017  2  ! The specified record was not found.  This condition indicates an internal
;  989          2018  2  ! logic error.
;  990          2019  2  !
;  991          2020  2  SIGNAL(JBC$_INVBLOCK OR STS$K_SEVERE, 1, .RECNO);
;  992          2021  1  END;



                                    001C 00000          .ENTRY  REWRITE_RECORD, Save R2,R3,R4                  ; 1921
                 54 00000000'   EF   9E 00002           MOVAB   QUEUE_RAB+4, R4                                 ;
          FCAC   C4         04   AC   D0 00009           MOVL    RECNO, DIAG_TRACE+20                            ; 1957
                          FD08   C4   D6 0000F           INCL    DIAG_COUNT+16                                   ; 1958
                 53       FF14   C4   9E 00013           MOVAB   BCB_ACTIVE_LIST, PREV                           ; 1959
                                 63   D5 00018 1$:       TSTL    (PREV)                                          ; 1960
                                 61   13 0001A           BEQL    6$                                              ;
                 52              63   D0 0001C           MOVL    (PREV), BCB                                     ; 1962
```

```
                  04    AC        04    A2  D1 0001F          CMPL    4(BCB), RECNO                              ; 1968
                                        57  1A 00024          BGTRU   6$                                         ;
                                        50  12 00026          BNEQ    5$                                         ; 1978
                  2C    A4        04    A2  9E 00028          MOVAB   4(BCB), QUEUE_RAB+48                       ; 1984
                  24    A4        0C    A2  D0 0002D          MOVL    12(BCB), QUEUE_RAB+40                      ; 1985
          09      FE98  C4              03  E1 00032          BBC     #3, FLAGS, 2$                              ; 1986
                        64 00100010     8F  D0 00038          MOVL    #1048592, QUEUE_RAB+4                      ; 1987
                                        03  11 0003F          BRB     3$                                         ;
                        64              10  D0 00041 2$:      MOVL    #16, QUEUE_RAB+4                           ; 1988
                                FCFC    C4  D6 00044 3$:      INCL    DIAG_COUNT+4                               ; 1989
                                  FC    A4  9F 00048          PUSHAB  QUEUE_RAB                                  ; 1990
             00000000G  00              01  FB 0004B          CALLS   #1, SYS$PUT                                ;
                        13              50  E8 00052          BLBS    R0, 4$                                     ;
                                  FC    A4  9F 00055          PUSHAB  QUEUE_RAB                                  ; 1992
                                  AC    A4  9F 00058          PUSHAB  QUEUE_FAB                                  ;
                            000410D4    8F  DD 0005B          PUSHL   #266452                                    ; 1993
             00000000G  EF              03  FB 00061          CALLS   #3, SIGNAL_FILE_ERROR                      ;
                                  08    A2  D7 00068 4$:      DECL    8(BCB)                                     ; 2000
                                        22  12 0006B          BNEQ    7$                                         ; 2001
                        63              62  D0 0006D          MOVL    (BCB), (PREV)                              ; 2004
                                        52  DD 00070          PUSHL   BCB                                        ; 2005
                  FBB6  CF              01  FB 00072          CALLS   #1, DEALLOCATE_BUFFER                      ;
                                            04 00077          RET                                                ; 1980
                        53              52  D0 00078 5$:      MOVL    BCB, PREV                                  ; 2013
                                        9B  11 0007B          BRB     1$                                         ; 1960
                                  04    AC  DD 0007D 6$:      PUSHL   RECNO                                      ; 2020
                                        01  DD 00080          PUSHL   #1                                         ;
                            0004841C    8F  DD 00082          PUSHL   #295964                                    ;
             00000000G  00              03  FB 00088          CALLS   #3, LIB$SIGNAL                             ;
                                            04 0008F 7$:      RET                                                ; 2021

; Routine Size:  144 bytes,    Routine Base:  CODE + 0485
```

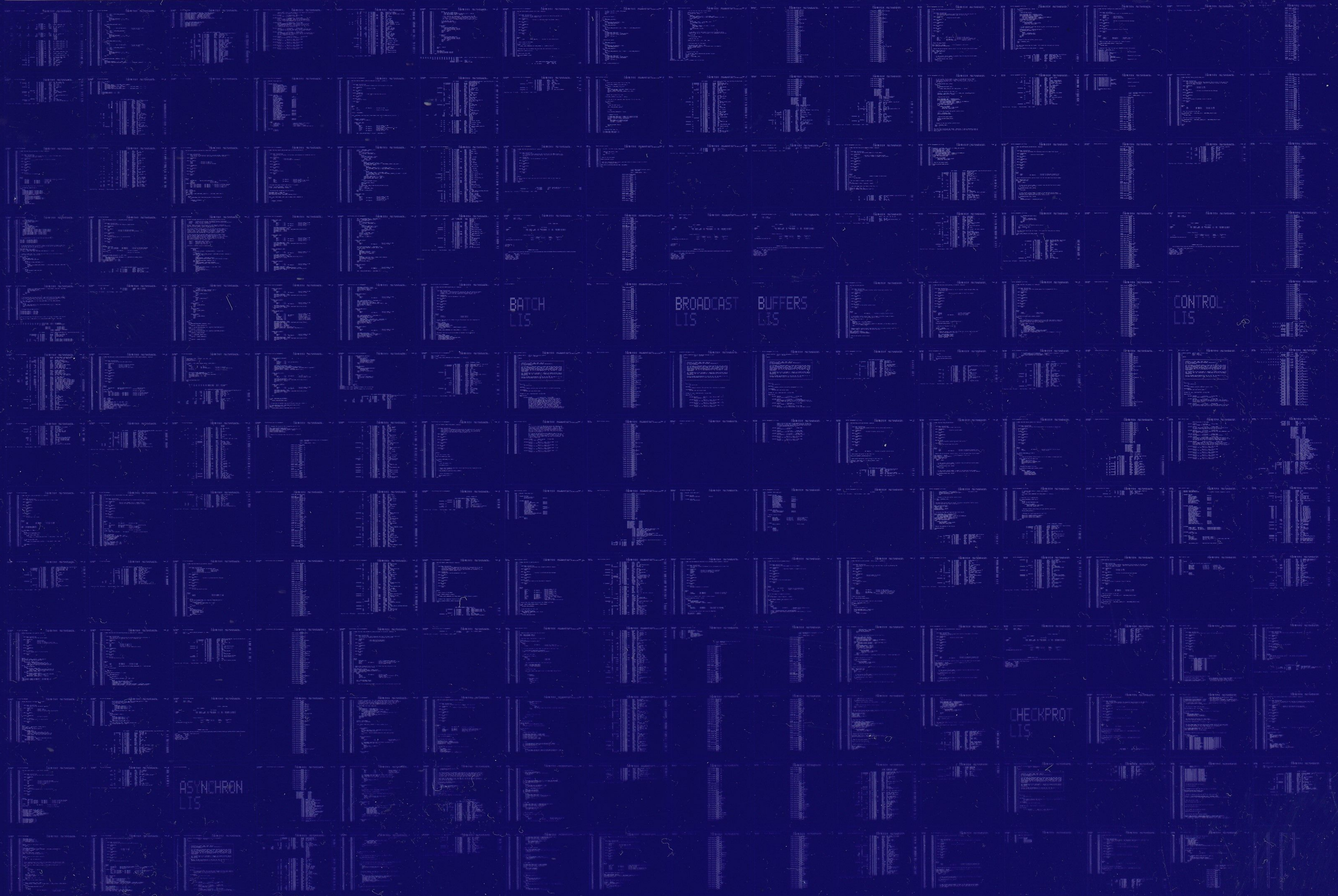
```
1 GLOBAL ROUTINE RELEASE_RECORD(RECNO): NOVALUE=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine decreases the reference count of a specified record.
1 !
1 ! INPUT PARAMETERS:
1 !       RECNO           - Relative record number.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 LOCAL
2       PREV,                           ! Pointer to previous forward link
2       BCB:            REF BBLOCK;     ! Pointer to buffer control block
2
2
2 ! Search the active buffer control block list for the specified record number.
2 !
2 PREV = BCB_ACTIVE_LIST;
2 WHILE ..PREV NEQ 0 DO
3     BEGIN
3     BCB = ..PREV;
3
3
3       ! If the current record number is greater than the desired record number,
3       ! the record is not in the list.
3       !
3     IF .BCB[BCB_RECNO] GTRU .RECNO
3     THEN
3       EXITLOOP;
3
3
3       ! If the current record number is equal to the desired record number,
3       ! decrease the reference count on the record.  If the reference count has
3       ! fallen to zero, deallocate the buffer control block.
3       !
3     IF .BCB[BCB_RECNO] EQLU .RECNO
3     THEN
4       BEGIN
4       BCB[BCB_REFCOUNT] = .BCB[BCB_REFCOUNT] - 1;
```

```
; 1051          2079 4              IF .BCB[BCB_REFCOUNT] EQL 0
; 1052          2080 4              THEN
; 1053          2081 5                  BEGIN
; 1054          2082 5                  .PREV = .BCB[BCB_LINK];
; 1055          2083 5                  DEALLOCATE_BUFFER(.BCB);
; 1056          2084 4                  END;
; 1057          2085 4              RETURN;
; 1058          2086 3              END;
; 1059          2087 3
; 1060          2088 3
; 1061          2089 3          ! Advance to the next buffer control block.
; 1062          2090 3          !
; 1063          2091 3          PREV = BCB[BCB_LINK];
; 1064          2092 2          END;
; 1065          2093 2
; 1066          2094 2
; 1067          2095 2  ! The specified record was not found.  This condition indicates an internal
; 1068          2096 2  ! logic error.
; 1069          2097 2  !
; 1070          2098 2  SIGNAL(JBC$_INVBLOCK OR STS$K_SEVERE, 1, .RECNO);
; 1071          2099 1  END;



                                           000C 00000          .ENTRY  RELEASE_RECORD, Save R2,R3                              ; 2022
                      53 00000000'     EF   9E 00002          MOVAB   BCB_ACTIVE_LIST, PREV                                   ; 2057
                                       63   D5 00009 1$:      TSTL    (PREV)                                                  ; 2058
                                       21   13 0000B          BEQL    3$                                                      ;
                      52               63   D0 0000D          MOVL    (PREV), BCB                                             ; 2060
                04    AC         04    A2   D1 00010          CMPL    4(BCB), RECNO                                           ; 2066
                                       17   1A 00015          BGTRU   3$                                                      ;
                                       10   12 00017          BNEQ    2$                                                      ; 2075
                                 08    A2   D7 00019          DECL    8(BCB)                                                  ; 2078
                                       22   12 0001C          BNEQ    4$                                                      ; 2079
                      63               62   D0 0001E          MOVL    (BCB), (PREV)                                           ; 2082
                                       52   DD 00021          PUSHL   BCB                                                     ; 2083
              FB75    CF               01   FB 00023          CALLS   #1, DEALLOCATE_BUFFER                                   ;
                                            04 00028          RET                                                             ; 2077
                      53               52   D0 00029 2$:      MOVL    BCB, PREV                                               ; 2091
                                       DB   11 0002C          BRB     1$                                                      ; 2058
                                 04    AC   DD 0002E 3$:      PUSHL   RECNO                                                   ; 2098
                                       01   DD 00031          PUSHL   #1                                                      ;
                          0004841C     8F   DD 00033          PUSHL   #295964                                                 ;
          00000000G   00               03   FB 00039          CALLS   #3, LIB$SIGNAL                                          ;
                                            04 00040 4$:      RET                                                             ; 2099

; Routine Size:  65 bytes,    Routine Base:  CODE + 0515
; 1073          2100   1 GLOBAL ROUTINE LOCK_QUEUE_FILE: NOVALUE=
; 1074          2101   1
; 1075          2102   1 !++
; 1076          2103   1 !
; 1077          2104   1 ! FUNCTIONAL DESCRIPTION:
; 1078          2105   1 !       This routine locks the system queue file in preparation for a
; 1079          2106   1 !       transaction.
; 1080          2107   1 !
; 1081          2108   1 ! INPUT PARAMETERS:
; 1082          2109   1 !       NONE
; 1083          2110   1 !
; 1084          2111   1 ! IMPLICIT INPUTS:
; 1085          2112   1 !       NONE
; 1086          2113   1 !
; 1087          2114   1 ! OUTPUT PARAMETERS:
; 1088          2115   1 !       NONE
; 1089          2116   1 !
; 1090          2117   1 ! IMPLICIT OUTPUTS:
; 1091          2118   1 !       NONE
; 1092          2119   1 !
; 1093          2120   1 ! ROUTINE VALUE:
; 1094          2121   1 !       NONE
; 1095          2122   1 !
; 1096          2123   1 ! SIDE EFFECTS:
; 1097          2124   1 !       NONE
; 1098          2125   1 !
; 1099          2126   1 !--
; 1100          2127   1
; 1101          2128   2 BEGIN
; 1102          2129   2 LOCAL
; 1103          2130   2         COUNT,                          ! Loop counter
; 1104          2131   2         STATUS;                         ! Status return
; 1105          2132   2
; 1106          2133   2
; 1107          2134   2 IF TESTBITCS(FLAGS[FLAGS_V_QUEUE_LOCKED])
; 1108          2135   2 THEN
; 1109          2136   3     BEGIN
; 1110          2137   3
; 1111          2138   3     ! Retry enqueue service on deadlock error up to 20 times before signalling
; 1112          2139   3     ! this error condition to prevent aborting the job controller on a false
; 1113          2140   3     ! deadlock indication from the lock manager.
; 1114          2141   3     !
; 1115          2142   3     INCR COUNT FROM 1 TO 20 DO
; 1116          2143   4         BEGIN
; 1117        P 2144   4         STATUS = $ENQW(
; 1118        P 2145   4             EFN=JBC$K_SYNC_EFN,
; 1119        P 2146   4             LKMODE=LCK$K_EXMODE,
; 1120        P 2147   4             LKSB=QUEUE_LOCK_LKSB,
; 1121          2148   4             FLAGS=LCK$M_CONVERT);
; 1122          2149   4         IF .STATUS THEN STATUS = .QUEUE_LOCK_LKSB[0];
; 1123          2150   4         IF .STATUS NEQ SS$_DEADLOCK THEN EXITLOOP;
; 1124          2151   4         DIAG_TRACE[11] = .DIAG_TRACE[11] + 1;    ! ***** diagnostic info *****
; 1125          2152   3         END;
; 1126          2153   3
; 1127          2154   3     IF NOT .STATUS
; 1128          2155   3     THEN
; 1129          2156   4         BEGIN
; 1130          2157  4         IF .FLAGS[FLAGS_V_CS_QF_DEADLOCK]       ! Conditionally bugcheck cluster
; 1131          2158  4             THEN FLAGS[FLAGS_V_CLUSTER_SCRAM] = TRUE;
; 1132          2159  4         SIGNAL(JBC$_COMREMJBC OR STS$K_ERROR, 0, .STATUS);
; 1133          2160  3         END;
; 1134          2161  2     END;
; 1135          2162  1 END;



                                                                        .EXTRN  SYS$ENQW

                                                     000C 00000         .ENTRY  LOCK_QUEUE_FILE, Save R2,R3                  ; 2100
                                        53 00000000' EF 9E 00002         MOVAB   FLAGS, R3                                    ;
                         51             63          00 E2 00009         BBSS    #0, FLAGS, 5$                                ; 2134
                                        52          01 D0 0000D         MOVL    #1, COUNT                                    ; 2142
                                                    7E 7C 00010 1$:     CLRQ    -(SP)                                        ; 2148
                                                    7E 7C 00012         CLRQ    -(SP)                                        ;
                                                    7E 7C 00014         CLRQ    -(SP)                                        ;
                                        7E          02 7D 00016         MOVQ    #2, -(SP)                                    ;
                                              00B0  C3 9F 00019         PUSHAB  QUEUE_LOCK_LKSB                              ;
                                                    05 DD 0001D         PUSHL   #5                                           ;
                                                    01 DD 0001F         PUSHL   #1                                           ;
                            00000000G   00          0B FB 00021         CALLS   #11, SYS$ENQW                                ;
                                        05          50 E9 00028         BLBC    STATUS, 2$                                   ; 2149
                                        50    00B0  C3 3C 0002B         MOVZWL  QUEUE_LOCK_LKSB, STATUS                      ;
                            00000E0A    8F          50 D1 00030 2$:     CMPL    STATUS, #3594                                ; 2150
                                                    08 12 00037         BNEQ    3$                                           ;
                                              FE2C  C3 D6 00039         INCL    DIAG_TRACE+44                                ; 2151
                         CF             52          14 F3 0003D         AOBLEQ  #20, COUNT, 1$                               ; 2142
                                        1A          50 E8 00041 3$:     BLBS    STATUS, 5$                                   ; 2154
                         04       02    A3          04 E1 00044         BBC     #4, FLAGS+2, 4$                              ; 2157
                                  02    A3          02 88 00049         BISB2   #2, FLAGS+2                                  ; 2158
                                                    50 DD 0004D 4$:     PUSHL   STATUS                                       ; 2159
                                                    7E D4 0004F         CLRL    -(SP)                                        ;
                                          00048412  8F DD 00051         PUSHL   #295954                                      ;
                            00000000G   00          03 FB 00057         CALLS   #3, LIB$SIGNAL                               ;
                                                       04 0005E 5$:     RET                                                  ; 2162

; Routine Size:  95 bytes,    Routine Base:  CODE + 0556
; 1137          2163  1 GLOBAL ROUTINE UNLOCK_QUEUE_FILE: NOVALUE=
; 1138          2164  1
; 1139          2165  1 !++
; 1140          2166  1 !
; 1141          2167  1 ! FUNCTIONAL DESCRIPTION:
; 1142          2168  1 !       This routine unlocks the system queue file following a transaction.
; 1143          2169  1 !
; 1144          2170  1 ! INPUT PARAMETERS:
; 1145          2171  1 !       NONE
; 1146          2172  1 !
; 1147          2173  1 ! IMPLICIT INPUTS:
; 1148          2174  1 !       NONE
; 1149          2175  1 !
; 1150          2176  1 ! OUTPUT PARAMETERS:
; 1151          2177  1 !       NONE
; 1152          2178  1 !
; 1153          2179  1 ! IMPLICIT OUTPUTS:
; 1154          2180  1 !       NONE
; 1155          2181  1 !
; 1156          2182  1 ! ROUTINE VALUE:
; 1157          2183  1 !       NONE
; 1158          2184  1 !
; 1159          2185  1 ! SIDE EFFECTS:
; 1160          2186  1 !       NONE
; 1161          2187  1 !
; 1162          2188  1 !--
; 1163          2189  1
; 1164          2190  2 BEGIN
; 1165          2191  2 LOCAL
; 1166          2192  2         BCB:            REF BBLOCK,     ! Pointer to buffer control block
; 1167          2193  2         STATUS;                         ! Status return
; 1168          2194  2
; 1169          2195  2
; 1170          2196  2 IF .QUEUE_FAB[FAB$W_IFI] NEQ 0
; 1171          2197  2 THEN
; 1172          2198  3     BEGIN
; 1173          2199  3
; 1174          2200  3     ! If the active buffer control block list is not empty, release each one.
; 1175          2201  3     !
; 1176          2202  3     WHILE .BCB_ACTIVE_LIST NEQ 0 DO
; 1177          2203  4         BEGIN
; 1178          2204  4         BCB = .BCB_ACTIVE_LIST;
; 1179          2205  4         BCB_ACTIVE_LIST = .BCB[BCB_LINK];
; 1180          2206  4         DEALLOCATE_BUFFER(.BCB);
; 1181          2207  3         END;
; 1182          2208  3
; 1183          2209  3
; 1184          2210  3     ! Release the queue file lock.
; 1185          2211  3     !
; 1186          2212  3     IF TESTBITSC(FLAGS[FLAGS_V_QUEUE_LOCKED])
; 1187          2213  3     THEN
; 1188          2214  4         BEGIN
; 1189        P 2215  4         STATUS = $ENQW(
; 1190        P 2216  4             EFN=JBC$K_SYNC_EFN,
; 1191        P 2217  4             LKMODE=LCK$K_NLMODE,
; 1192        P 2218  4             LKSB=QUEUE_LOCK_LKSB,
; 1193          2219  4             FLAGS=LCK$M_CONVERT);
; 1194          2220  4             IF .STATUS THEN STATUS = .QUEUE_LOCK_LKSB[0];
; 1195          2221  4             IF NOT .STATUS
; 1196          2222  4             THEN
; 1197          2223  4                 SIGNAL(JBC$_COMREMJBC OR STS$K_ERROR, 0, .STATUS);
; 1198          2224  3             END;
; 1199          2225  3
; 1200          2226  3
; 1201          2227  3         ! Close the file if appropriate.
; 1202          2228  3         !
; 1203          2229  3         IF .QUEUE_REFERENCE_COUNT EQL 0
; 1204          2230  3         THEN
; 1205          2231  4             BEGIN
; 1206          2232  5             IF NOT $CLOSE(FAB=QUEUE_FAB)
; 1207          2233  4             THEN
; 1208          2234  4                 SIGNAL_FILE_ERROR(
; 1209          2235  4                     JBC$_CLOSEOUT OR STS$K_ERROR,
; 1210          2236  4                     QUEUE_FAB, QUEUE_FAB);
; 1211          2237  4
; 1212          2238  4
; 1213          2239  4             ! Release context associated with open $GETQUI operations.
; 1214          2240  4             !
; 1215          2241  4             DELETE_OPEN_GETQUIS();
; 1216          2242  4
; 1217          2243  4
; 1218          2244  4             ! Release all locks acquired for the file.
; 1219          2245  4             !
; 1220          2246  4             $DEQ(LKID=.(REMOTE_REQUEST_LKSB[2]));
; 1221          2247  4             $DEQ(LKID=.(QUEUE_FILE_LKSB[2]));
; 1222          2248  4             $DEQ(LKID=.(QUEUE_LOCK_LKSB[2]));
; 1223          2249  3             END;
; 1224          2250  2         END;
; 1225          2251  1 END;


                                                                   .EXTRN  SYS$CLOSE, SYS$DEQ

                                             001C 00000            .ENTRY  UNLOCK_QUEUE_FILE, Save R2,R3,R4          ; 2163
                          54 00000000G    00   9E 00002            MOVAB   SYS$DEQ, R4                               ;
                          53 00000000'    EF   9E 00009            MOVAB   QUEUE_FAB, R3                             ;
                                  02      A3   B5 00010            TSTW    QUEUE_FAB+2                               ; 2196
                                          01   12 00013            BNEQ    1$                                        ;
                                               04 00015            RET                                               ;
                          50        FF68  C3   D0 00016 1$:        MOVL    BCB_ACTIVE_LIST, R0                       ; 2202
                                          11   13 0001B            BEQL    2$                                        ;
                          52              50   D0 0001D            MOVL    R0, BCB                                   ; 2204
                   FF68   C3              62   D0 00020            MOVL    (BCB), BCB_ACTIVE_LIST                    ; 2205
                                          52   DD 00025            PUSHL   BCB                                       ; 2206
                   FAD1   CF              01   FB 00027            CALLS   #1, DEALLOCATE_BUFFER                     ;
                                          E8   11 0002C            BRB     1$                                        ; 2202
              31   FEEC   C3              00   E5 0002E 2$:        BBCC    #0, FLAGS, 4$                             ; 2212
                                          7E   7C 00034            CLRQ    -(SP)                                     ; 2219
                                          7E   7C 00036            CLRQ    -(SP)                                     ;
                                          7E   7C 00038            CLRQ    -(SP)                                     ;
                          7E              02   7D 0003A            MOVQ    #2, -(SP)                                 ;
                                  9C      A3   9F 0003D            PUSHAB  QUEUE_LOCK_LKSB                           ;
                          7E              01   7D 00040            MOVQ    #1, -(SP)                                 ;
```

```
          00000000G  00            0B  FB 00043        CALLS   #11, SYS$ENQW                              ;
                     07            50  E9 0004A        BLBC    STATUS, 3$                                 ; 2220
                     50        9C  A3  3C 0004D        MOVZWL  QUEUE_LOCK_LKSB, STATUS                    ;
                     11            50  E8 00051        BLBS    STATUS, 4$                                 ; 2221
                                   50  DD 00054 3$:    PUSHL   STATUS                                     ; 2223
                                   7E  D4 00056        CLRL    -(SP)                                      ;
                         00048412  8F  DD 00058        PUSHL   #295954                                    ;
          00000000G  00            03  FB 0005E        CALLS   #3, LIB$SIGNAL                             ;
                             FF24  C3  D5 00065 4$:    TSTL    QUEUE_REFERENCE_COUNT                      ; 2229
                                   42  12 00069        BNEQ    6$                                         ;
                                   53  DD 0006B        PUSHL   R3                                         ; 2232
          00000000G  00            01  FB 0006D        CALLS   #1, SYS$CLOSE                              ;
                     11            50  E8 00074        BLBS    R0, 5$                                     ;
                                   53  DD 00077        PUSHL   R3                                         ; 2234
                                   53  DD 00079        PUSHL   R3                                         ;
                         0004105A  8F  DD 0007B        PUSHL   #266330                                    ; 2235
          00000000G  EF            03  FB 00081        CALLS   #3, SIGNAL_FILE_ERROR                      ;
          00000000G  EF            00  FB 00088 5$:    CALLS   #0, DELETE_OPEN_GETQUIS                    ; 2241
                                   7E  7C 0008F        CLRQ    -(SP)                                      ; 2246
                                   7E  D4 00091        CLRL    -(SP)                                      ;
                               90  A3  DD 00093        PUSHL   REMOTE_REQUEST_LKSB+4                      ;
                     64            04  FB 00096        CALLS   #4, SYS$DEQ                                ;
                                   7E  7C 00099        CLRQ    -(SP)                                      ; 2247
                                   7E  D4 0009B        CLRL    -(SP)                                      ;
                               98  A3  DD 0009D        PUSHL   QUEUE_FILE_LKSB+4                          ;
                     64            04  FB 000A0        CALLS   #4, SYS$DEQ                                ;
                                   7E  7C 000A3        CLRQ    -(SP)                                      ; 2248
                                   7E  D4 000A5        CLRL    -(SP)                                      ;
                               A0  A3  DD 000A7        PUSHL   QUEUE_LOCK_LKSB+4                          ;
                     64            04  FB 000AA        CALLS   #4, SYS$DEQ                                ;
                                       04 000AD 6$:    RET                                                ; 2251

; Routine Size:  174 bytes,    Routine Base:  CODE + 05B5
; 1227          2252  1 END
; 1228          2253  0 ELUDOM



                                                                  .EXTRN  LIB$SIGNAL

;                                        PSECT SUMMARY
;
;       Name                         Bytes                         Attributes
;
;   COMMON                              5024  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(2)
;   CODE                                1635  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                                 Library Statistics
;
;                                                 -------- Symbols --------        Pages        Processing
;       File                                      Total    Loaded   Percent       Mapped       Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1                 18619        63         0        1000          00:01.4





;                                         COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:BUFFERS/OBJ=OBJ$:BUFFERS MSRC$:BUFFERS/UPDATE=(ENH$:BUFFERS)

; Size:           1635 code + 5024 data bytes
; Run Time:          00:28.5
; Elapsed Time:      01:44.6
; Lines/CPU Min:       4746
; Lexemes/CPU-Min: 40055
; Memory Used:   275 pages
; Compilation Complete
```

0191 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



BATCH
LIS

BROADCAST
LIS

BUFFERS
LIS

CONTROL
LIS

CHECKPROT
LIS

ASYNCHRON
LIS